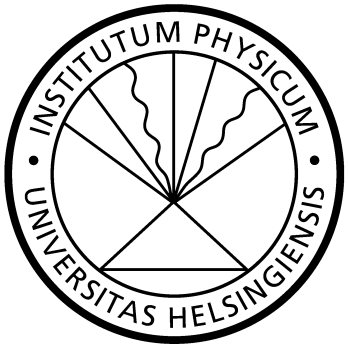


Master's thesis
Computational materials research

Bayesian optimization structure search

Henri Paulamäki
2019

Supervisor: Patrick Rinke
Examiners: Patrick Rinke
Kai Nordlund

HELSINGIN YLIOPISTO
FYSIIKAN LAITOS

PL 64 (Gustaf Hällströmin katu 2)
00014 Helsingin yliopisto

HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI
MATEMAATTIS-LUONNONTIETEELLINEN TIEDEKUNTA
MATEMATISK-NATURVETENSKAPLIGA FAKULTETEN
FACULTY OF SCIENCE

Tiedekunta – Fakultet – Faculty Matemaattis-luonnontieteellinen tiedekunta	**Koulutusohjelma – Utbildningsprogram – Degree programme** Materiaalitutkimuksen maisteriohjelma	
Tekijä – Författare – Author Henri Paulamäki		
Työn nimi – Arbetets titel – Title Bayesian optimization structure search		
Työn laji – Arbetets art – Level Pro gradu	**Aika – Datum – Month and year** 11/2019	**Sivumäärä – Sidoantal – Number of pages** 65

Tiivistelmä – Referat – Abstract

Tailoring a hybrid surface or any complex material to have functional properties that meet the needs of an advanced device or drug requires knowledge and control of the atomic level structure of the material. The atomistic configuration can often be the decisive factor in whether the device works as intended, because the materials' macroscopic properties - such as electrical and thermal conductivity - stem from the atomic level. However, such systems are difficult to study experimentally and have so far been infeasible to study computationally due to costly simulations.

I describe the theory and practical implementation of a 'building block'-based Bayesian Optimization Structure Search (BOSS) method to efficiently address heterogeneous interface optimization problems. This machine learning method is based on accelerating the identification of a material's energy landscape with respect to the number of quantum mechanical (QM) simulations executed. The acceleration is realized by applying likelihood-free Bayesian inference scheme to evolve a Gaussian process (GP) surrogate model of the target landscape. During this active learning, various atomic configurations are iteratively sampled by running static QM simulations. An approximation of using chemical building blocks reduces the search phase space to manageable dimensions. This way the most favored structures can be located with as little computation as possible. Thus it is feasible to do structure search with large simulation cells, while still maintaining high chemical accuracy. The BOSS method was implemented as a python code called aalto-boss between 2016-2019, where I was the main author in co-operation with Milica Todorović and Patrick Rinke.

I conducted a dimensional scaling study using analytic functions, which quantified the scaling of BOSS efficiency for fundamentally different functions when dimension increases. The results revealed the target function's derivative's important role to the optimization efficiency. The outcome will help people with choosing the simulation variables so that they are efficient to optimize, as well as help them estimate roughly how many BOSS iterations are potentially needed until convergence. The predictive efficiency and accuracy of BOSS was showcased in the conformer search of the alanine dipeptide molecule. The two most stable conformers and the characteristic 2D potential energy map was found with greatly reduced effort compared to alternative methods. The value of BOSS in novel materials research was showcased in the surface adsorption study of bifenyldicarboxylic acid on CoO thin film using DFT simulations. We found two adsorption configurations which had a lower energy than previous calculations and approximately supported the experimental data on the system.

The three applications showed that BOSS can significantly reduce the computational load of atomistic structure search while maintaining predictive accuracy. It allows material scientists to study novel materials more efficiently, and thus help tailor the materials' properties to better suit the needs of modern devices.

Avainsanat – Nyckelord – Keywords
Bayesian optimization, Structure search, Hybrid surfaces, DFT

Säilytyspaikka – Förvaringställe – Where deposited
E-thesis

Muita tietoja – Övriga uppgifter – Additional information

Contents

1 Introduction

State of the art technological devices are small and combine different kinds of materials – like molecules and crystals – to bring out the best in each. Often the materials are stacked as thin layers, which means that there are many hybrid surfaces featuring e.g. an organic-inorganic interface. To mention a few, hybrid surfaces appear in electronic components like semiconductor junctions, integrated optics, selective coatings, gas sensors and solar cells. The union of materials in an organic solar cell for example, allows efficient light adsorption of organic molecules to be combined with good electric conductivity and durability of metals to capture and transport solar energy.

Tailoring a hybrid surface or any complex material to have functional properties that meet the needs of an advanced device or drug requires knowledge and control of the atomic level structure of the material. The atomistic configuration can often be the decisive factor in whether the device works as intended, because the materials' macroscopic properties – such as electrical and thermal conductivity – stem from the atomic level. Finding the most favorable atomic configurations which lead to desired macroscopic properties, has so far been computationally infeasible. This is due to the high computational cost of the quantum mechanical simulation methods used to accurately calculate properties of atomistic configurations. There exist approximate simulation methods of lesser computational cost (e.g. force fields and interatomic potentials) but so far their results are accurate enough for only certain types of materials at a time. They can for example be parametrized to accurately describe proteins, but yield unrealistic results for metal oxides. For this reason they cannot be used for accurately studying e.g. organic/inorganic interfaces in heterogeneous devices – such as organic solar cells. To experimentally measure the properties of the device is relatively easy, but to determine the underlying atomistic structure is very difficult for nanoscale interfaces. This is especially difficult if the interface features soft materials – like organic molecule layers – as the interference of the measuring device can easily change the structure.

In this work I describe the theory and practical implementation of a 'building block'-based Bayesian Optimization Structure Search (BOSS) method to address heterogeneous interface optimization problems. This machine learning method's feasibility and accuracy are demonstrated both in a molecular conformer search study and a surface adsorption study. Additionally the dimensional scaling of the method is benchmarked against simple analytic functions. The objective of this work is to offer a practical method for optimizing e.g. hybrid surface structures, and to show that it is computationally feasible, while still using accurate quantum mechanical simulations to calculate material properties.

The BOSS method is based on accelerating the identification of a material's energy landscape with respect to the number of quantum mechanical (QM) simulations executed. The acceleration is realized by applying likelihood-free Bayesian inference scheme to evolve a Gaussian process (GP) surrogate model of the target landscape. During this active machine learning, various atomic configurations are iteratively sampled by running static QM simulations, with the objective to locate the most favored structures with as little computation as possible. This enables large simulation cells to be used in the structure search, while still maintaining high chemical accuracy and feasible computational load.

Alternative methods to conduct efficient structure searches include methods like biased molecular dynamics[8] (MD) umbrella sampling[16], reverse integration[10] and basin-hopping[24, 23] for sampling the configurational phase space combined with analytic models (such as the reaction path Hamiltonian approximation[23]) to interpolate between sampled data. While carefully directed, the simulations conducted in these methods are measured in nano or picoseconds of MD or number of structural relaxations. Thus the number of point energy evaluations in the phase space is in the thousands or higher depending on the time step. For this reason the ability of the BOSS method to cut the number of static simulations needed down to a few hundreds makes significantly larger atomic configurations feasible to globally optimize. Additionally the GP model of the target energy landscape can be data mined after a BOSS search to predict several local minima as well as minimum energy paths and barriers between them. GPs have been used before in atomistic structure search, but just in GP regression[9] (without BO) which is local optimization – i.e. it only finds the nearest minimum structure from the starting configuration.

The BOSS method was implemented as a python code called `aalto-boss` between 2016-2019, where I was the main author in co-operation with Patrick Rinke, Milica Todorović, Harshit Mahapatra, Ester Koistinen (all from Aalto university) and Ville Parkkinen (University of Helsinki). There was also collaboration with computer science experts from both Aalto university and University of Helsinki. This code was used in the demostrative BOSS studies in section 4, and its development was a major part of this work. The code combines the BO algorithm with GPs (GPy[18] library) and total energy simulations into a single user-friendly program. It is intended to be usable not only for machine learning experts but for any material scientist with a complex optimization problem. This is possible through a set of good default parameters for the machine learning variables as well as support for any choice of material simulator and computing environment.

I present in this work results of three applications of the BOSS method: a dimensional scaling study with simple analytic functions (section 4.1), conformer search problem us-

ing gradient information to accelerate the BOSS method, and a novel material adsorption study. The scaling study compares how efficiently different kinds of simple but fundamentally different functions are optimized using BOSS as the dimension of the problem increases. The findings can be used to estimate how long a time a certain complicated optimization might take. In the conformer search of alanine dipeptide, the predictive power of BOSS is demonstrated and efficiency compared to other methods. Simultaneously the effect of adding function gradients information to BOSS is measured and another scaling estimate provided. In the last result I tackle a recent problem: the adsorption of bifenyldicarboxylicacid on CoO thin film. In this study, the value of BOSS in novel materials research is showcased, as lower energy adsorption configurations than before are found after optimizing the complex surface adsorption system. These results fulfill the objectives of this work by showcasing the feasibility and accuracy of the BOSS method.

2 Theory

2.1 Overview of BOSS

Bayesian optimization structure search (BOSS) is a method combining machine learning and atomistic structure search. The core idea is to use Bayesian optimization (section 2.3) to find structures of the system which optimize a target property, while making as few total energy simulations (section 2.4) as possible. A third element of equal importance in BOSS is the choice of suitable building blocks or simulation variables (section 2.2) $\mathbf{x} \in \chi$, which define the coordinates of the configurational phase space. In this work the target property to optimize, $f(\mathbf{x})$, is the potential energy of the atomistic system.

The Bayesian optimization (BO) algorithm iteratively refines a surrogate model of an objective function (see Figure 1), which in this case is the potential energy $E_p(\mathbf{x})$ as a function of the simulation variables $\mathbf{x}$. BO uses acquisition functions (section 2.3.1) to control the sampling of the space χ such, that the target property could be globally optimized fast. There exist many total energy simulation methods, which can make such a potential energy evaluation for a given structure. The problem is that the methods that are accurate, are computationally expensive to execute for large systems. On the other hand, the methods that are cheap to calculate are inaccurate e.g. for hybrid materials. For that reason the capability of accurate predictions with small data, makes BOSS a computationally feasible method to study larger systems than before, whilst using accurate *ab initio* simulations methods.

The BO algorithm in BOSS employs Bayesian subjective probability view to update a

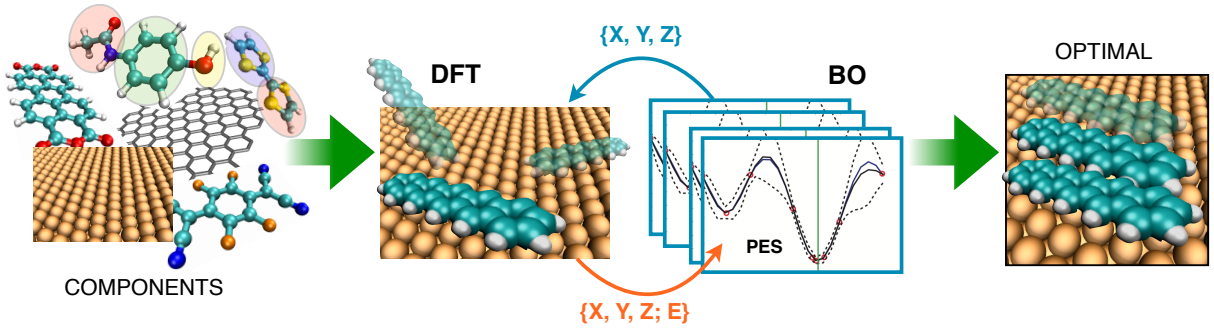


Figure 1: Schematic of the BOSS method's core idea. The building block components identified from atomic systems are arranged energetically optimally by iteratively sampling the phase space with total energy simulations and simultaneously refining a surrogate model of the potential energy surface (PES). The model indicates where to sample next, while the atomistic simulations append the information set of the true PES until the model converges and shows the optimal configuration. Reproduced wtih permission from [4].

Gaussian process (GP) prior prediction into a posterior GP prediction based on collected data $(\mathbf{x}, E_p(\mathbf{x}))$. The surrogate model includes both a prediction (posterior mean) and an uncertainty (posterior variance) inferred over the entire potential energy landscape. The model then indicates the most informative next point to sample via the use of acquisition functions.

The use of Gaussian process models makes the results of a BOSS search versatile to use. Even though the primary goal often is to find the global minimum energy structure, the GP model contains plenty of information from other parts of the energy landscape (see Figures 1 and 5). Especially if there exist several minima of similar depth, the complete set of local minima becomes more important than just the global minimum. Additionally, transition pathways between minima and energy barriers on those pathways provide insight into the dynamic properties of the system. To ensure accuracy of model prediction away from the global minimum, it is necessary to run BOSS for more iterations after the global minimum prediction has converged. Nevertheless the efficiency of BOSS in terms of computational cost of total energy simulations is very high.

In the remainder of this section I will explain in detail the three main aspects of BOSS: the building block principle of choosing the phase space variables, the BO algorithm and total energy simulations. Moreover, I will cover technical features of the GO surrogate model, several ways to accelerate BOSS optimization and the methods to quantify and track BOSS efficiency. Note that the central terminology that is presented in the following subsections and repeatedly used in this work, is collected in a short glossary in section 6.

2.2 Choosing simulation variables

In principle an atomic structure with N atoms has $3N$ degrees of freedom (DOF): three spatial coordinates per atom $\mathbf{x} = \sum_i^N (p_{xi}, p_{yi}, p_{zi})$. Considering N to be at least in the hundreds (where it is for instance for heterogeneous interface structures), this number of variables is too much to construct an accurate model with few enough evaluations of the structures' energy. With how few evaluations the total energy can be optimized in BOSS depends strongly on how many DOF does one consider the simulations to have. One may choose the used DOFs in any way one likes, as long as they uniquely define an atomic structure which can be simulated to acquire its energy (see Figure 2). Fortunately, not nearly all of the available $\approx 3N$ DOFs need to be optimized, because their local minimum energy structures are known from previous research. In other words, when a group of atoms is forms a certain known structure, one can often safely approximate that that structure doesn't change significantly and fix it in place. For instance the structure of a benzene ring in an organic molecule can often be approximated to stay in its symmetric hexagonal minimum structure. This means that the problem reduces to optimizing how the ring as a rigid unit positions itself with respect to other parts of the molecule, instead of optimizing the positions of all the six carbons individually. In this section I describe a so called "building block" approach for selecting the most important DOFs as the simulation variables, which are optimized and modelled with respect to the energy of the structure they correspond to. The building blocks here mean stable groups of atoms that don't likely change their internal configuration.

In order to optimize sufficiently large structures, it is necessary to keep the dimension of the problem low regardless of the large number of atoms. In many atomic systems one can identify certain "building blocks", which have a previously known internal structure, but could be arranged in various ways with respect to each other. The known internal structure could be for example the known rigid structure of an adsorbate molecule in a surface adsorption system, or the stable structure of a functional group of an organic chain-like molecule in conformer search. In other words we approximate that certain bond lengths and angles are fixed, which is a valid assumption for many local structures with an energetically deep minimum configuration. The exact positions of each atom can of course be obtained later by relaxing the forces in the optimal structure found by BOSS.

In the building block approach, the DOFs of the movement of the building blocks are taken to be the simulation variables. In practise this means, that for the mentioned adsorption system, the translational (x, y, z) and rotational (α, β, γ) coordinates of each of the adsorbate molecules span the space of the potential energy surface (PES), which we are trying to model and optimize (see Figure 2 middle illustration). This way one

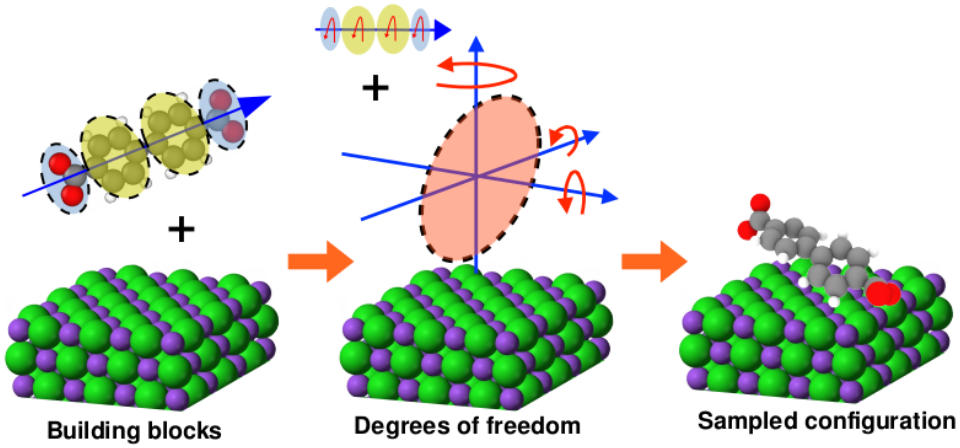


Figure 2: The building blocks (left) consist of the rigid parts in the system plus possible internal variables, like here the benzene rings and carboxyl groups of bifenyldicarboxylic acid molecule. The degrees of freedom (middle) are the three translational and three rotational coordinates of the adsorbate plus the internal simulation variables of the molecule. Here the molecule's internal variables are the rotations of the building blocks around the molecular axis. Each set of values for the degrees of freedom defines a unique configuration (right) in the phase space, which can be sampled for its energy using total energy simulation methods.

could perform structure search with two adsorbates using 12 dimensions to find out in what angle the molecules adsorb, and how they pile up with respect to each other (see Figure 3).

In case of the organic chain, the movement of the functional groups – i.e. our building blocks – with respect to each other can be realized by selecting the dihedral angles of the chain as the variables. These correspond to rotating the backbone bonds of the chain around their longitudinal axis. The dihedral angles have also been found by nonlinear manifold learning techniques[8] to be the simulation variables (or order parameters), which affect the energy of these structures the most. Using these variables, the molecule is free to visit all relevant conformers, but the problem dimension is reduced to the number of bonds in the backbone plus bonds off the backbone to functional groups.

The building block approach can be applied to various atomic systems, and it enables a global structure search of large simulation cells. However, one should carefully base the choice of used building blocks on physical and chemical knowledge. Not always can an adsorbate molecule for example be approximated to be rigidly fixed to its gas phase equilibrium structure. While this is often a good approximation when the molecule is far away from the surface, the interaction with the surface could cause molecular distortions when it is near. Some distortion is always to be expected, but if it is considered energetically

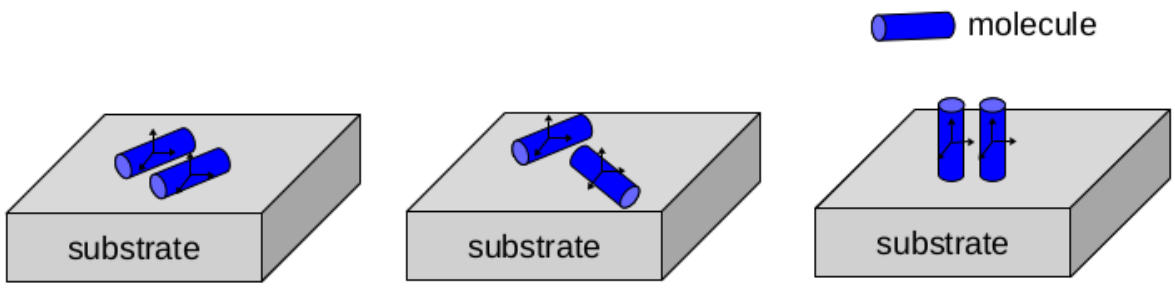


Figure 3: Schematic of a few different ways molecules could stack up on a substrate. From left to right a planar parallel, planar anti-parallel and vertical parallel two molecule configurations are illustrated.

significant, it should be taken into account as an additional variable.

While the chosen "building block" variables have to allow the structure to visit all relevant configurations, they should avoid the possibility for uninteresting and problematic structures. When given an atomic structure with atoms too close (≈ 0.5Å) to each other, simulators will either give an extremely high energy or the calculation will fail completely. For this reason, for example, the z coordinate of an adsorbate molecule should have its lower bound set so that the molecule is never placed too close to the surface when BOSS probes for different structures. In a case of two adsorbate molecules, any clashes between the two molecules should also be avoided. This could be done for example by using the distance between the molecules as one building block variable, and by setting that variable's lower limit appropriately.

Symmetries of the chosen variables are known to significantly reduce the number of BOSS iterations until the model is converged. For this reason choosing variables which have some symmetry, is advisable whenever possible. However, the symmetries must be acknowledged by the user, and that extra information needs to be explicitly put into the BOSS search. The most important case is if a variable is periodic (e.g. a rotational variable spanning an entire 360° spin or a translational variable across a unit cell). The periodicity can be incorporated into the GP model covariance function. The details of how this is done are discussed in Section 2.3. Another case of speeding up BOSS with the use of variable symmetries, is when variables have a mirror symmetry. In this case a data acquisition made in one location, must by symmetry have the same value at some other location as well. Thus, several data points could be added to the data set after doing only one evaluation (static simulation).

After minimum energy structures have been found efficiently using BOSS, it is of course possible to allow the found structures to relax completely. Thus the known structures

selected as the building blocks can relax all the atomic positions and lower the energy even further. So the building block strategy helps one do the optimization in manageable dimensions but doesn't restrict the structures to be found.

2.3 Bayesian optimization

Bayesian optimization (BO) is an machine learning algorithm for globally optimizing an N-dimensional objective function $f(\mathbf{x}), \mathbf{x} \in \chi$ with as few evaluations as possible. In BO the objective function is assumed unknown and frequently modelled by a Gaussian process (GP). The GP is fitted on training data $(\mathbf{X}, \mathbf{y})$ acquired iteratively by evaluating the objective function $\mathbf{y} = f(\mathbf{X})$. In addition to a prediction, the GP model provides an uncertainty measure for each point in the domain space χ, and it is used to direct the data acquisition so that always the most informative point would be selected. As more and more data is collected the GP model converges to the objective function and loses its uncertainty. This happens first in the minimum areas of objective function value and eventually in all space. In addition to just global optimization the GP model constructed in BO can be analyzed further to find other minima as well as paths and barriers between them.

The BO algorithm is represented by the following step-by-step pseudocode:

1. Acquire initial training data to start with $D = (\mathbf{X}, \mathbf{y})$.
2. Fit a Gaussian process on the training data.
3. Optimize the GP model hyperparameters (optional).
4. If a stopping condition is met, finish. Otherwise continue.
5. Minimize the acquisition function to select next sampling point $\mathbf{x_{next}}$.
6. Evaluate $f(\mathbf{x_{next}})$ and append it to training data set D.
7. Return to step 2.

Figure 4: BO algorithm

The algorithm starts by evaluating the objective function a few times to collect a small initial data set (step 1), which serves as input to construct the first GP fit (step 2). From that point on, steps 2-7 are repeated until a predefined stopping condition is met in step 4. The usual condition in global optimization is that either a preset maximum number of iterations is reached or the global minimum prediction has converged – i.e. the GP model no longer changes the location or value of its prediction of the global minimum. In BO, considerable effort is put into carefully selecting the locations where to acquire

new data points on each iteration. An acquisition function is formed based on the latest GP model fit and minimized to indicate the most informative next sampling point $\mathbf{x_{next}}$. Note that this sampling method does not require any human intuition as would be typical for traditional structure search.

The remainder of this subsection will focus on the details of two important parts of the BO algorithm: the acquisition of new data and the GP model. Also the choice and optimization of the GP's covariance function is explained.

2.3.1 Acquisitions

The data acquisitions in the BO algorithm are selected with special care because each one of them is computationally very costly. The entire N-dimensional phase space needs to be explored to ensure global optimization and at the same time the precise location of the minimum should be found accurately. For this reason random or uniform sampling would be too inefficient in accurately locating the minimum, but using MD would mean too little exploration of less frequent structures.

In the BO algorithm, the GP posterior mean and variance (denoted $\boldsymbol{\mu}(\mathbf{x}_*)$ and $\boldsymbol{\Sigma}(\mathbf{x}_*)$ and defined better later) are used to determine the next sampling point to acquire, such that it would be as informative as possible. This means that the acquisitions are selected in such locations, that the GP model will fit/find the global minimum as fast as possible. The strategy is to take the next data acquisition at the maximum of model uncertainty $\mathbf{x}_{next} = \text{argmax}(\boldsymbol{\Sigma}(\mathbf{x}_*))$. This results in phase space exploration, as the next query point is always chosen at regions of the phase space far away from where there are previous training data points. Collecting data in this manner necessarily makes the model mean converge to the objective function with an arbitrary accuracy as more and more is collected. However, this kind of sampling alone will make the convergence of global minimum prediction very slow.

A smart and flexible solution for picking the data acquisition locations is the use of an acquisition function. It is a function calculated based on the posterior mean (prediction) and variance (uncertainty) of the current GP, and its global minimum indicates the most informative next sampling point.

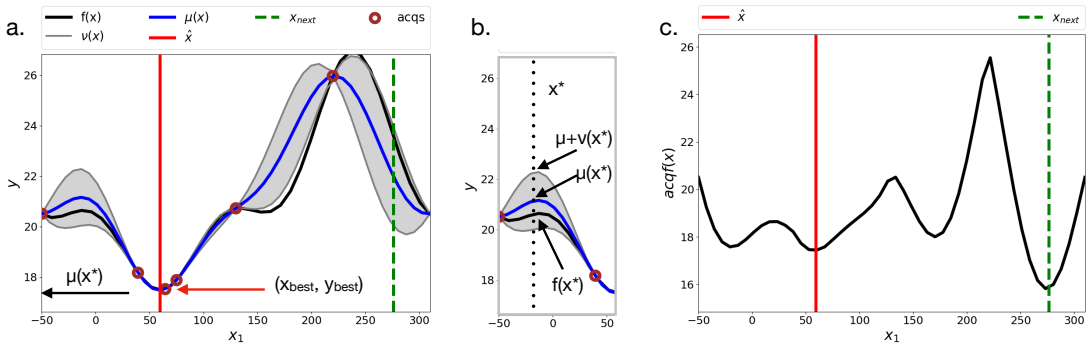


Figure 5: Illustration of the important quantities of a GP surrogate model (a.-b.) and the corresponding acquisition function (c.). In subfigure a. the x-axis is an arbitrary simulation variable and the y-axis shows the energy. The two curves drawn are the true PES $f(x*)$ (black line) and the GP model $\mu(\mathbf{x}*)$ (blue line) fitted to the data (red circles) sampled from the true PES. The uncertainty $\nu(x*)$ of the GP model is drawn so that the higher bound of the grey shaded area is $\mu(x*) + \nu(x*)$ and the lower bound is $\mu(x*) - \nu(x*)$. Subfigure (c.) shows the LCB acquisition function (see Equation 1) calculated based on the GP in (a.). The red vertical line shows the currect model prediction – i.e. the global minimum $\text{argmin}(\mu(x*)) = \hat{x}$ – while the green dashed vertical line shows the next sampling location, which is the minimum of the acquitision function $x_{next} = \text{argmin}(acqf(x*))$. It can be seen that as the GP already has the true PES minimum accurately predicted, the next sampling point is directed to an area of larger uncertainty to explore the PES. Reproduced wtih permission from [4].

Though there are different acquisition functions one could use, the most relevant ones for atomistic structure search try to balance between exploration and exploitation (sampling where the minimum is predicted to be). This ensures that minima of the objective function are found with as few function evaluations as possible, but still the entire space is somewhat explored in order to not miss the global minimum. A simple but effective acquisition function is the lower confidence bound (LCB)

$$\text{LCB}(\mathbf{x}) = \boldsymbol{\mu}(\mathbf{x}) - \eta\boldsymbol{\Sigma}(\mathbf{x}), \mathbf{x} \in \chi. \tag{1}$$

In equation 1 the first $\boldsymbol{\mu}(\mathbf{x})$ and second $-\eta\boldsymbol{\Sigma}(\mathbf{x})$ term correspond to exploration and exploitation, respectively, as we are minimizing model prediction and minimizing the negative of uncertainty. The positive coefficient η is the balancing parameter between exploration and exploitation. According to Corander and Gutmann[11] the value of η should be chosen to increase as a function of BO iteration and search dimensionality d to avoid getting stuck at a local minimum:

$$\eta = \sqrt{2\log[i^{(\frac{d}{2}+2)}\pi^2/0.3]}. \tag{2}$$

The i is the BO iteration number, which is the training data set size so far apart from the initial points. Using such an increasing exploration coefficient, helps to prevent acquisition function from getting stuck acquiring data at a few places only. Another common acquisition function is the expected improvement (EI):

$$\mathrm{EI}(\mathbf{x}) = E[f(\mathbf{x}_{best}) - \mathrm{N}(\boldsymbol{\mu}(\mathbf{x}), \boldsymbol{\Sigma}(\mathbf{x}))], \tag{3}$$

where $\mathbf{x}_{best}$ is the best (lowest) point in training data. EI balances between exploration and exploitation too, but this trade-off is slightly different than for LCB (Eq. 1) and also less transparent from its functional form.

Despite using an acquisition function, it is possible that in some cases its minimum points to the same place where there already is a data point. If one would then sample at this place again, the GP nor the acquisition function would change at all, and the same point would neither get chosen as the next sampling point again. To overcome this problem, there are modifications one can use on top of the acquisition function. An effective one, which was also used in the results of this work, is to define conditions under which to switch to pure exploration. In other words, if the conditions are met, one changes the acquisition function to just $-\boldsymbol{\Sigma}(\mathbf{x})$ for one iteration. A condition for switching to pure exploration which we have found to work well is the following:

$$\text{if} \quad \boldsymbol{\Sigma}(\mathbf{x}_{next}) < \epsilon \quad \rightarrow \quad \text{pure exploration.} \tag{4}$$

ϵ represents a small tolerance constant. If the GP posterior variance (uncertainty) at the proposed sampling point is very low, it indicates that the region is already "known" quite well, and we wish not to waste any more data acquisitions there. Such a modification can help the acquisition function get unstuck and continue to operate effectively again.

Next, I discuss the role of the initial data points. They are the training data points used to fit the first GP model, before the acquisition function is used to pick the rest of the training data. In which locations and how many initial points should one then select? The minimum is two data points, because a GP fit on a single point or no points is always just flat. However, when we evaluate the initial points, we are doing it "blind" – that is, without acquisition function to sample in informative locations. For this reason one is lead to think that the minimum of two initial points would be the best choice. On the other hand the GP fit on only very few points is vulnerable to over-fitting or over-smoothing when the model is optimized. We concluded it is still best to keep the number of initial points low as we can then use the acquisition function early on. The initial points we pick are spread out uniformly in the space according to the quasi-random Sobol sequence[21], because exploration of space is a safe choice when no information about possible minimum location is available.

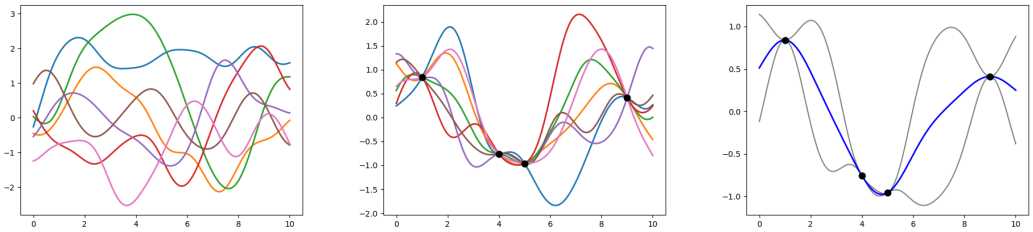
2.3.2 Gaussian process, kernel and hyperparameters

In BOSS the value of the target property is predicted in all of the phase space by a Gaussian process (GP). A GP surrogate model is the joint Gaussian distribution of random variables indexed by a continuous space, which makes up a normally distributed prediction for all points in the continuous phase space. This subsection explains the basic theory of GPs with the perspective of how it is used in BOSS method.

Before any training data the GP model starts out as a flat prior distribution with zero mean and covariance function $\mathbf{K}$:

$$f \sim \mathrm{N}(\mathbf{0}, \mathbf{K}). \tag{5}$$

The prior is a distribution over functions (see FIG 6a) in the space χ of the objective function. It is the initial belief of how the objective function behaves. We choose a flat prior, because generally we do not have a valid guess for the exact shape of the function. However, in a structure search problem we do often have a some guess for how fast the target function can vary and what kind of a range of values the function can have. This information we encode in the covariance function $\mathbf{K}$. It describes the similarity between points in χ and as such expects certain characteristic properties for the fitted GP. The central role of the covariance function will be discussed in more detail later in this subsection.



(a) Prior distribution (b) Posterior distribution (c) Posterior mean and variance

Figure 6: The principle of a Gaussian process (GP) fit. The prior distribution (a) over functions is completely random and has mean zero. It is updated to a posterior distribution (b) given data evaluated from the objective function. The GP model (c) is considered the mean and variance of the posterior distribution, which are interpreted as the prediction and its uncertainty.

When one has evaluated the objective function to obtain training data $(\mathbf{X}, \mathbf{y})$, one can update the GP prior into a posterior distribution (see FIG 6b). The functions making up to the posterior all fit through the training data points within a certain noise (which is

very small). From the posterior distribution, one can make predictions about the value of the objective function at any test point $\mathbf{x}_* \in \chi$. This is done by calculating the mean and variance of the posterior at the test point. The mean is given by

$$\boldsymbol{\mu}(\mathbf{x}_*) \quad = \mathbf{K}_*(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y} \tag{6}$$

and variance by

$$\boldsymbol{\Sigma}(\mathbf{x}_*) \quad = \mathbf{K}_{**} - \mathbf{K}_*(\mathbf{K} + \sigma_n^2\mathbf{I})^{-1}\mathbf{K}_*^{\mathrm{T}}. \tag{7}$$

Above we have denoted $\mathbf{K}_{**} = \mathbf{K}(\mathbf{x}_*, \mathbf{x}_*)$, $\mathbf{K}_* = \mathbf{K}(\mathbf{X}, \mathbf{x}_*)$ and $\mathbf{K} = \mathbf{K}(\mathbf{X}, \mathbf{X})$, which are the covariance matrices internally for the test point, between the training data and test point, and internally for the training data. The error term $\sigma_n^2\mathbf{I}$ is due to that we allow the objective function evaluations to have some small Gaussian noise with standard deviation σ_n. While the energy calculations in the structure search are noiseless, the small noise term is there to ensure numerical stability. The relevant mathematics for deriving equations 6 and 7 can be found in literature[17]. The posterior mean $\boldsymbol{\mu}(\mathbf{x}_*)$ and variance $\boldsymbol{\Sigma}(\mathbf{x}_*)$ are interpreted as the prediction for objective function values at the test point $f(\mathbf{x}_*)$ and its uncertainty (see FIG 6c).

The GP model can be understood so that it provides a one dimensional Gaussian distribution as the prediction for the objective function value $f(\mathbf{x}_*)$ at all points $\mathbf{x}_* \in \chi$. As illustrated in Figure 6, near the training data $(\mathbf{X}, \mathbf{y})$ the GP model uncertainty vanishes[1], as all the posterior distribution's functions must fit through those points. Conversely, far away from the training data points, the GP model has the most uncertainty, because many different values for the objective function could be possible there.

As differentiation is a linear operation, the derivative of a GP is also a GP [22]. This makes incorporating gradient observations – i.e. force information – into the model easy. As

$$\mathrm{cov}\left(\frac{\partial f^{(i)}}{\partial x_g^{(i)}}, f^{(j)}\right) \quad = \quad \frac{\partial}{\partial x_g^{(i)}}\mathrm{cov}\big(f^{(i)}, f^{(j)}\big), \text{ and} \tag{8}$$

$$\mathrm{cov}\left(\frac{\partial f^{(i)}}{\partial x_g^{(i)}}, \frac{\partial f^{(j)}}{\partial x_h^{(j)}}\right) \quad = \quad \frac{\partial^2}{\partial x_g^{(i)}\partial x_h^{(j)}}\mathrm{cov}\big(f^{(i)}, f^{(j)}\big), \tag{9}$$

the covariance matrices $\mathbf{K}$ in Equations 6 and 7 (in section 2.3) can be extended to include gradient observations. This extra information significantly improves the GP fit accuracy with small data (see Figure 11 for illustration).

[1]Technically it doesn't go exactly to zero, because of the very small noise we are considering the training data to have.

The covariance function $\mathbf{K}(\mathbf{x}, \mathbf{x}')$ of the GP model determines the aspects of the surrogate model fitted on the data. The kernel inside the covariance function has continuous parameters called hyperparameters which can be optimized to balance model selection so, that overfitting and underfitting would be minimal. The covariance function answers the question: if we know the objective function value at $\mathbf{x}$, how much can we know about it at point $\mathbf{x}'$. The answer is affected by at least three distinct features:

- length scale – how rapidly does the function change between minima and maxima?
- variance – how big is the range of function values $\mathbf{y}$?
- periodicity – is there a periodic boundary condition like $f(\mathbf{x}) = f(\mathbf{x} + \mathbf{T})$?

These are all general features of the objective function, which can be encoded in the covariance function $\mathbf{K}$. If these features are known correctly for the objective function, the GP can predict very accurately even with a small number of training data points. Fortunately in structure search applications one does often know what kind of length scales, variances and periodicity to expect for the variables based on physical and chemical intuition. For example a coordinate translating an adsorbate atom across a flat surface will have about as many minima and maxima as there are atoms on the surface along the direction of the translation. Moreover a variable corresponding to rotating an angle in a molecule obviously must have a 2π period unless another symmetry reduces the period to be smaller. Rough guesses for the objective function's general features are enough, because in the BO algorithm they are learned as more training data is collected. Should the initial guesses be completely wrong, the BO will eventually fit the GP correctly, because exploration of the search space is guaranteed and the GP hyperparameters are updated regularly. However, this may require many more training data points compared to if the initial guesses were in the correct order of magnitude.

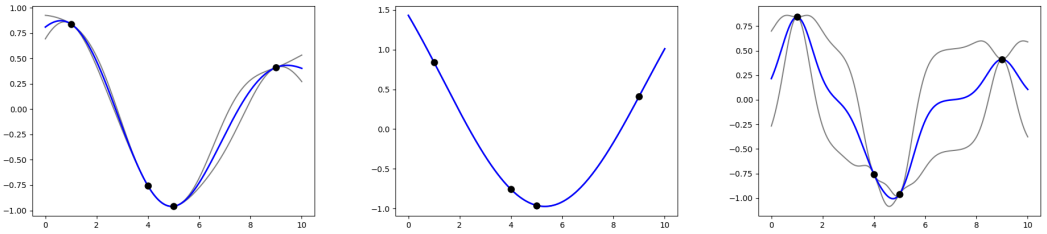
The quantification of the general features comes through the mathematical form of the kernel. There are various possibilities for the form, but I will present two very common ones here. Radial basis function[2] (RBF) is written as:

$$\mathbf{K}_{RBF}(\mathbf{x}, \mathbf{x}') = \sigma^2 exp\left(-\frac{1}{2}\sum_{i=1}^{dim}\frac{(x_i - x_i')^2}{\ell_i^2}\right). \tag{10}$$

The parameters σ and ℓ correspond to the variance and length scale respectively. They, along with any other possible parameters of the kernel, are called hyperparameters. The length scale hyperparameter ℓ works so, that the smaller its value is, the faster the function

[2]Also known as the squared exponential

is expected to fluctuate. An illustration related to this, is shown in figure 7. It is also noteworthy that length scale can and often will be different in each dimension (kernel is non-isotropic), whereas variance is only a single value for the entire function.



(a) Optimized length scale (b) Triple length scale (c) One third length scale

Figure 7: An illustration of the effect of kernel length scale hyperparameter on the Gaussian process (GP) fit on training data. The underlying objective function in this example is $f(x) = \sin(x)$. (a) shows the GP prediction and uncertainty to the black training data points with optimized value for the length scale. (b) shows the GP fit with a length scale value three times the optimized one. The too large length scale results in over-smoothing, which visually looks like a function with as little complexity as possible would be fitted to the data. The GP fit with one third of the optimized length scale value is shown in (c). Too small length scale causes variance between training data points to be exaggerated and unnecessary fluctuations to appear to the prediction.

RBF is a non-periodic kernel, but its periodic counterpart is the Standard periodic:

$$\mathbf{K}_{StdP}(\mathbf{x}, \mathbf{x}') = \sigma^2 \exp\left(-\frac{1}{2}\sum_{i=1}^{dim}\frac{\sin(\frac{2\pi}{T_i}(x_i - x_i'))^2}{\ell_i^2}\right). \tag{11}$$

In equation 11 we have a third type of hyperparameter T_i. It is the periods of the function separately for each dimension.

The correct periods of the variables are often known beforehand exactly, but the length scales and variance only roughly at best. The correct values are found in BO algorithm by optimizing the GP model several times while iteratively collecting more data. The GP is optimized by maximizing the marginal likelihood as a function of the hyperparameters. Marginal likelihood is a measure of how likely the GP with certain hyperparameter values is the best fit to the given set of training data. Hyperparameter optimization is a challenging multidimensional optimization problem itself, so it must be done carefully to get it right. With a small training data set, the correctly optimized hyperparameter values are often not the ones that later turn out to be the suitable values for an accurate GP fit. If there are no restrictions for the hyperparameters in the optimization, their

values typically vary a lot during the early iterations of the BO. The reason for this is that the optimization of hyperparameters itself is a multidimensional optimization problem, that can have very different solutions when the data set changes. The fluctuation of hyperparameter values can cause occasional bad GP model fits and as a result poor sampling choices indicated by the acquisition function. To overcome this problem and accelerate BO, one can set constraints or priors on the hyperparameters. Constraints mean hard boundaries below and over which the values cannot go when optimized. Priors on the other hand are probability distributions, which indicate the most likely values of the hyperparameter. Both kind of restrictions are based on the initial guesses for hyperparameters, but priors are a more flexible solution as they are more forgiving if the guesses were wrong. Even though the prior's mean is at the wrong location, it still allows other values far from the mean with non-zero probability. Additionally the contribution of the prior diminishes with the number of data points collected. As a restatement of what was concluded earlier in this section: if the hyperparameter values are far from optimal, BO will be less efficient, but the correct GP fit will eventually be always found when enough data has been collected.

2.4 Atomistic total energy simulations

To perform structure search on an atomistic scale, one needs a simulator to link an atomistic structure to an energy[6]. The law's of physics governing the world of atoms are encoded in a simulator. This make it possible to calculate the energy and forces in a given atomistic structure. Different simulators vary in their level of approximation regarding the laws and thereby in their accuracy and computational cost. Lightweight simulators – such as Lennard-Jones potential or AMBER[1] forcefield – include only classical electrostatics, van der Waals corrections and parameterized potentials, and are therefore very fast to simulate but inaccurate for other than certain near-ideal systems. On the other end of the range of simulators, quantum mechanics (QM) is taken into account in *ab initio* methods. The latter kind of simulators provide very accurate results for most systems, but require significantly more computational resources. Atomistic simulations are practically always run parallel on many computing cores in supercomputers, but the difference between classical and QM simulators can be the difference between hours to days or between days to weeks (classical being faster and QM slower) of supercomputer time depending on the simulated system. Due to this limitation, people studying large systems (larger than nanometer scale) and long timescale ($\gtrsim ps$) dynamic phenomena have so far had to resort to classical parameterized semi-empirical simulators. There are also systems, for which it has been hard to find an accurate simulator at all that would still

be computationally feasible. For example organic-inorganic interfaces – vital systems to understand for making heterogeneous devices of tailored functional properties – cannot be treated accurately by most parameterized potential simulators because of the diversity of different kinds of elements in the system. Parameterized potential simulators are known to be specialized to certain type of elements at a time. They can provide accurate results for example for metals, non-metals, oxides, organic or inorganic, but not for all at a time. On the other hand, the interfaces are too large for most high accuracy simulators to handle if traditional structure search methods are used and thus many simulations needed. The BOSS method enables the study of such systems with high accuracy simulators, by cutting down the number of simulator evaluations needed to locate the minimum energy structures.

Most atomistic simulation methods output many other quantities than just the total energy, such as for example charge distribution, band gap and electrical or thermal conductivity. After BOSS structure search based on energy evaluations, the resulting GP model could also be fitted to model the other properties as a function of the simulation variables.

To test the performance of BOSS method in section 4, AMBER[1] – the Assisted Model Building with Energy Refinement – force field was used as the simulator in search of conformers of alanine dipeptide molecule. AMBER belongs to the class of parameterized classical potentials simulators, and it works particularly well for organic molecules. With AMBER the energies are fast to calculate, making it feasible to conduct efficiency and scaling determination as described in section 3.4. The `Sander` forcefield (Amber16 manual section 18.1, [1]) that I used has the following form:

$$V(\mathbf{r}_i) = \sum_{bonds} k_b(l - l_0)^2 + \sum_{angles} k_\theta(\theta - \theta_0)^2 + \sum_{dihedrals} (V_n/2)[1 + \cos(n\phi - \delta)] + \sum_{i \neq j} \Big[(A_{ij}/r_{ij}^{12}) - (B_{ij}/r_{ij}^{6}) + (q_i q_j/r_{ij})\Big]. \quad (12)$$

In Equation 12 one can identify separate terms for the energy contribution of bond stretching, bond angle turning and dihedral angle (molecule's internal rotations) twisting with respect to their equilibrium values. The last term corresponds to van der Waals short range repulsion ($+r^{12}$) and long range attraction ($-r^6$) as well as the electrostatic coulomb potential between all pairs of atoms. The electric charges q, distances r, bond lengths l, angles θ and rotations ϕ are known from the given atomic structure, but the equilibrium values and spring constants are parameters. The parameterization is chosen so that the forcefield simulates the interactions between atoms as accurately as possible compared to QM simulations and experimental results. Good standard values for most atoms and

bonds are already determined (using more accurate methods) and freely available in the scientific community. A set of atoms, their positions and information about which ones are bonded, is all that is needed to compute the energy of the structure with a parameterized AMBER forcefield. The time taken for AMBER to acquire the total energy of a structure with a few tens of atoms in a serial calculation is of the order of 0.1 seconds on one standard desktop CPU. The time taken for the actual energy calculation is probably smaller, but a considerable part of the single point energy calculation goes to initialization and file operations.

In the results for bifenyldicarboxylic acid on CoO-Ir surface in section 4.3, VASP[5] code was used as the simulator. It implements **density functional theory**[14] (DFT), to calculate the ground state solution to the many-body Schrödinger equation. In DFT the Born-Oppenheimer approximation is employed to remove the nucleus-nucleus and nuclei kinetic energy terms from the equation. The rest is written in terms of the electronic density $[n]$, which greatly reduces the number of variables in the problem:

$$E[n] \quad = \quad T_e[n] + E_{ext}[n] + E_{Hartree}[n] + E_{xc}[n]. \tag{13}$$

In Equation 13 the known terms from the left are the kinetic energy of electrons, external electrostatic potential and the Hartree energy caused by interactions between static ions and their own electrons. The last term is the exchange correlation functional which is in practise approximated since it is unknown. The Hohenberg-Kohn-theorems[12] state that the electron density **n** which minimizes the energy functional (Equation 13), is the ground state electron density uniquely for the given external potential. As the potential is determined by the nuclei (elements and their positions in space), one can calculate the energy of an atomistic structure using DFT. While this is in principle an exact result, there are always approximations in the exchange correlation functional. PBE[13] is one commonly used functional which incorporates density gradients into E_{xc} on top of the simple local density approximation (LDA[15]). Minimizing the energy functional following the variational principle is significantly slower than calculating the energy using classical analytic formulas, which makes DFT a computationally much more costly atomistic simulator. On the other hand, DFT is based on a more fundamental physical description of interatomic energies, making it generally predict much more accurate results than classical potentials (unless the predicted structure is included in a classical potential code's fitting database – i.e. when the potential is specialized to model that type of structures).

3 BOSS code

3.1 Code implementation

The python implementation for BOSS method, which was used to make the results (section 4) in this master's thesis work, arose from a collaboration across research fields and years of work. The two parties first becoming aware of a possible joint interest (in 2015-16) were the Computational Electronic Structure Search (CEST) group of Aalto University Dept. of Applied physics and the Bayesian Statistics group at University of Helsinki, Dept. of Mathematics and Statistics. After the idea of applying Bayesian optimization (BO) to atomistic structure search was born, it started out by sharing an in-house Matlab BO code. It had been developed in the Bayesian statistics group to implement the BO algorithm as described in section 2.3, but not aimed at users outside their community nor easily ready to scale. Thus, plans for creating on open-source HPC-portable python code dedicated to BOSS started to form.

I started in CEST group as a summer student in June 2016 and begun working looking into ways for developing our own BOSS code. With the help of Patrick Rinke (professor and group leader), Milica Todorović (post-doc), Harshit Mahapatra (summer student 2017), Ester Koistinen (summer student 2018) and Ville Parkkinen (doctoral student) I developed a python code implementing the BOSS method. The code depends on the `GPy`-package[18] for Gaussian processes and the user for carrying out the total energy simulations of their choice. At early stages there were versions of the code which additionally dependended on the `GPYgradients`-package[19] – a version of `GPy` with gradient information at observations included in the GPs (see theory in section 2.3). In addition to testing and studying literature, the code development included consulting with both Jukka Corander (professor, Bayesian Statistics group, HU) and Aki Vehtari (professor, Probabilistic machine learning group, Aalto univ.) as well as their respective research groups. From my part the work on code development was done over the course of three years – in form of nine months of full-time and nine months of part-time work.

The BOSS code is written entirely in python for two reasons. Firstly it enables easy co-operation with the stable python package `GPy`, and secondly python is supported on many HPC facilities (in contradiction to Matlab), which are required for the total energy simulations that are a part of BOSS method. Note that most of the computational resources go to running the expensive atomistic simulations when using the BOSS code. For example in section 4.3 simulations take roughly 20 minutes each, while the other parts are some seconds per iteration. The execution on the BO side in python spends most of its time inverting the covariance matrices for GP predictions and optimizing

various analytical functions. Such frequent optimization sub-problems are minimization of the acquisition function and GP prediction on every iteration and the maximization of marginal likelihood in hyperparameter updates. A standard strategy of starting multiple local (LBFGS-B) optimizers from different points is taken in an effort to obtain reliable enough global optima. These bigger tasks on the python part of the code are carried out by calling functions from standard python libraries `NumPy` and `SciPy`, which are written mostly in `C` underneath.

When plugging an atomistic simulation code into BOSS, one only needs to worry about implementing a so called user function, which implements the DOFs (or building blocks) and returns the energy given a set of simulation variable values. This typically means rotating and translating atoms and molecules to build the queried structure, running the simulation in parallel and parsing the resulting energy. However, if gradient observations are used in the GP, the user function should return the gradient as well. This can be more difficult as most atomistic simulation codes output just the forces on each atom in the structure, and the gradient with respect to all simulation variables needs to be calculated from the forces. For example for a translation variable of one or more free atoms (e.g. translation of an entire molecule) the gradient is a sum $\nabla E = -\sum_i^N \vec{f}_i \cdot \hat{t}$, where $\vec{f}_i$ is the force on atom i and $\hat{t}$ is the unit vector in direction of the translation. For the alanine dipeptide result in section 4.2 I implemented the gradient calculation in case of a rotating functional group. In that case the gradient was $\nabla E = \sum_i^N \vec{r}_i \times \vec{f}_i - \sum_j^M \vec{r}_j \times \vec{f}_j$, where $\vec{f_{i(j)}}$ are the forces and $\vec{r_{i(j)}}$ the distance vectors of atoms $i(j)$ from the axis of rotation. i goes over the N atoms in the functional group while j goes over the M atoms in the rest of the molecule. While this worked, it felt at that time unnecessary to implement it for all the rotation variables in case of the fast classical simulator used in that study. So instead I ended up calculating the gradients simply by using the finite difference method $\partial_x E(x) = \frac{1}{2\epsilon}(E(x-\epsilon) + E(x+\epsilon))$.

The remainder of section 3 focuses on a simple example BOSS search and practical details of the method, that have been found while doing research using BOSS.

3.2 Example BOSS search

This section features a set-by-step example of a BOSS search in 1D. While the method has already been explained in full detail, this example serves as a practical demonstration of how it all works. This way the reader will be better equipped to understand what exactly has been done in the case studies of the results section 4. In this 1D demonstration I show the optimization of the function $f(x) = \sin(x) + \sin(2x)$. I chose this function as it's optimization is not trivial but can still be shown in a handful of iterations. The search

space is $x \in [-2, 2\pi - 2]$ – one full period $f(x)$. The periodic kernel from Equation 11 is used for the GP model and its hyperparameter values (excluding the period fixed to 2π) are updated on every iteration. The acquisition function used in this demo is the EI (expected improvement) from Eq. 3. Initial data set is constructed by making two acquisitions at the two first 1D Sobol sequence points [0.0] and [0.5] – that is, at the leftmost boundary and at the middle of the search space. The initial GP fitted on those two points is shown in Figure 8a as the model of iteration 0. So far the model predicts the minimum to be at the lower of the two initial points (at $x = -2$).

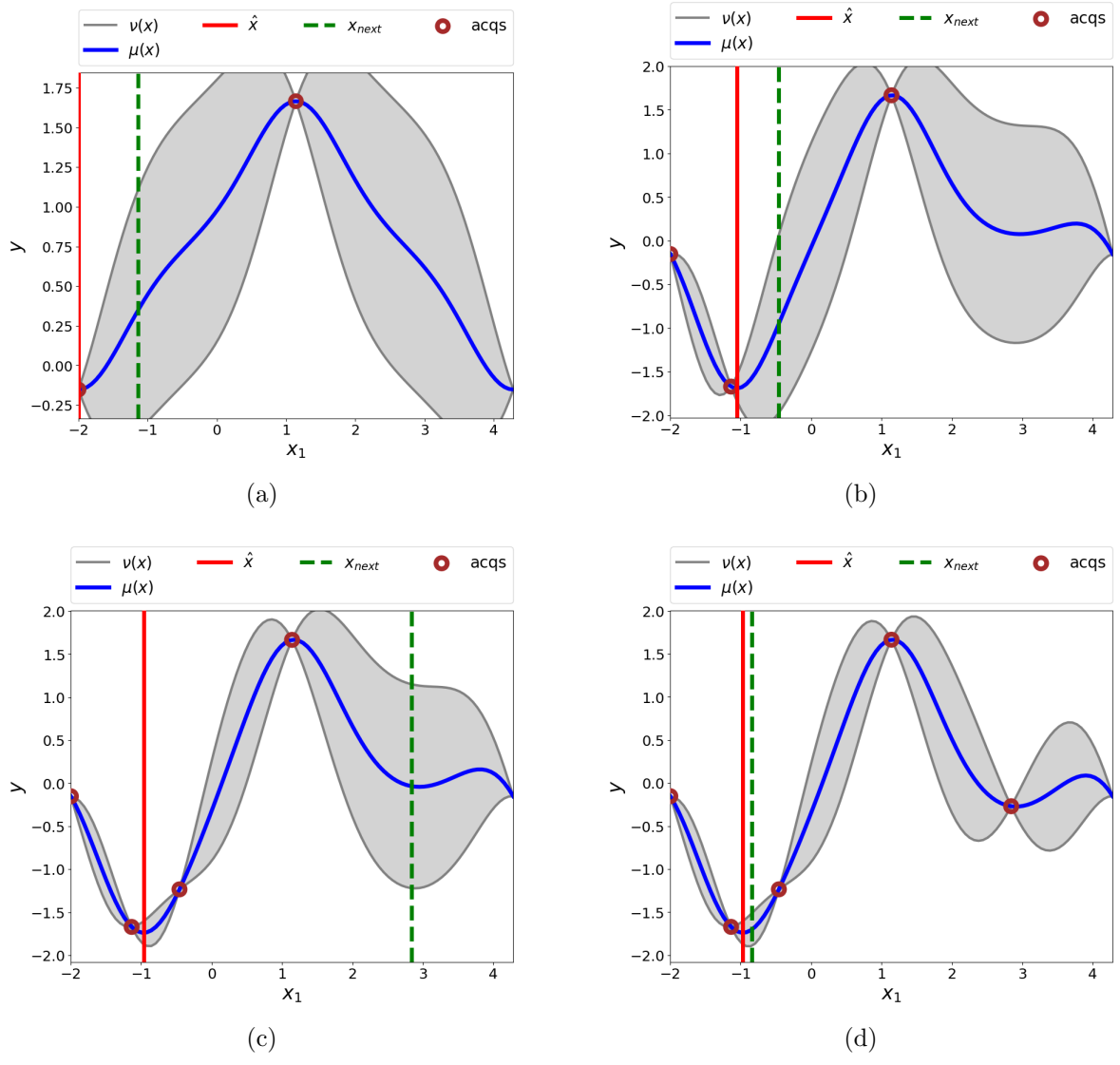


Figure 8: BOSS model evolution one acquisition at a time. Iterations 0-3 are shown in subfigures a-d with increasing amount of acquired data points (circles). The different lines are: blue - GP prediction, grey - GP uncertainty, red - predicted minimum and green - next acquisition.

The next point of acquisition is directed to the large uncertain area between the initial points but slightly closer to the lower one. Next we proceed one iteration and data acquisition at a time until the GP model converges.

The third data point revealed an even lower function value shifting the minimum prediction to the right (model in Figure 8b). Now that there is evidence of three points, the length scale indicated by the formed well enables the prediction of a second minimum on the right.

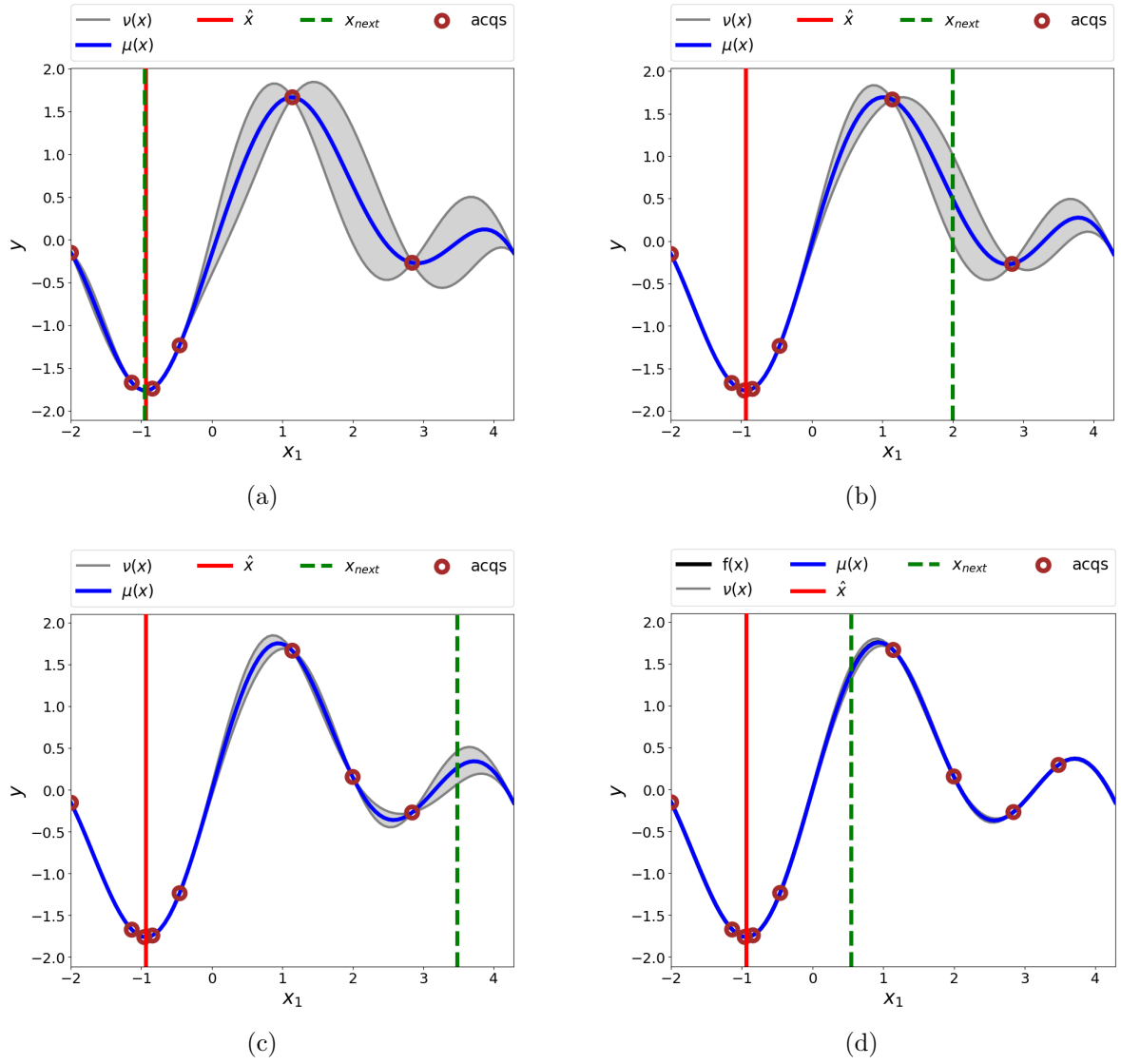


Figure 9: BOSS model evolution one acquisition at a time. Iterations 4-7 are shown in subfigures a-d with increasing amount of acquired data points (circles). The different lines are: blue - GP prediction, grey - GP uncertainty, red - predicted minimum and green - next acquisition.

The area around the second minimum prediction still has large uncertainty, but the next acquisition is nevertheless directed towards the area of moderate uncertainty right next to the current global minimum prediction. This displays the trade-off between uncertainty and low function value in the next acquisition selection using EI.

After evaluation of the fourth data point at $x \approx -0.5$ (Fig. 8c), the GP model has narrowed down the uncertainty near $x \approx -1$, where the global minimum is now predicted to be between the two closest data points in that area. The acquisition function now points to the bottom of the second predicted minimum on the right – a clearly exploratory step. That evaluation roughly agreed with the existing prediction at that point, thus not changing it by very much (see Fig. 8d). On the other hand the uncertainty around the second minimum was greatly reduced, making the acquisition function point back towards the global minimum.

Near the left-side minimum there still exists slightly uncertain areas between the data points, which is rooted out by taking two of the next acquisitions in that area in Figures 9a and 9b. After that evidence there is no uncertainty to be seen by eye anymore in the interval $x \in [-2, 0]$. From now on the global minimum at $x \approx -1$ is so accurately predicted that the acquisition function does not point extra evaluations to $x \in [-2, 0]$ for the rest of this BOSS search. Instead the remaining acquisitions (Figs 9c-d) are taken at uncertainty maxima on higher function value areas on the right side of the graphs. In other words the rest of the space is explored until the final model in Figure 9d has barely any uncertainty.

To ensure that the converged prediction is visually correct, the true function $f(x) = \sin(x) + \sin(2x)$ is plotted on the back ground in Figure 9d. It cannot be seen at all as the blue line marking the model prediction is so accurately on top of it. The exact value of the predicted global minimum in the GP model from BOSS code output is $(x = -0.93593, f = -1.76017)$. A quick analytic calculation:

$$\begin{aligned}
\frac{df}{dx} &= \cos(x) + 2\cos(2x) \quad = 0 && (14)\\
\cos(x) &= -2\cos(2x) && (15)\\
x &= 2\pi n \pm 2\tan^{-1}\left(\sqrt{6 \pm \sqrt{33}}\right), \quad n \in Z && (16)\\
\text{minimum} \quad f(2\pi n - 0.935929) &= -1.76017 && (17)
\end{aligned}$$

confirms that it is correct up to four decimals.

The GP model convergence can be seen visually very nicely here in 1D. However, generally in an N-dimensional optimization with only slices of the model visually available to see, it is better to look at the evolution of key quantities.

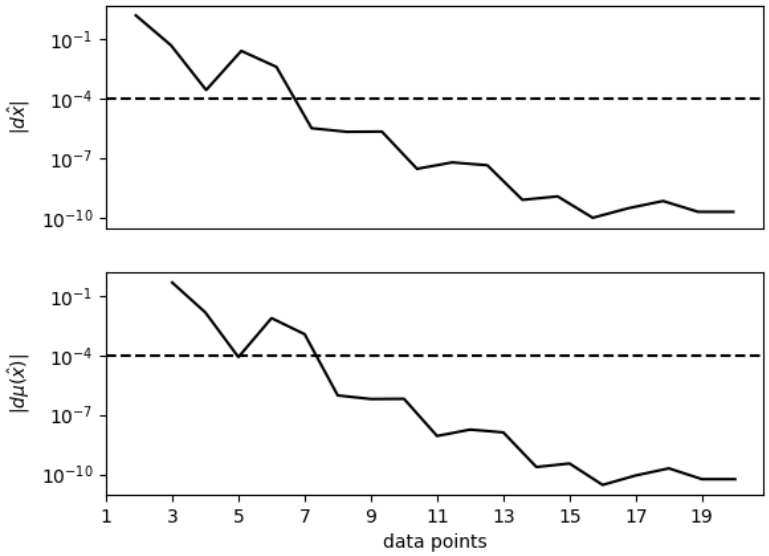


Figure 10: Evolution of the absolute change of the minimum prediction location $\hat{x}$ (top) and value $\mu(\hat{x})$ (bottom) in the example BOSS search. To make it a scale-independent measure, the $|d\mu(\hat{x})|$-value is additionally divided by the y-range, which is the difference between highest and lowest (so far) acquired data point. The dashed horizontal line marks the value 10^{-4}, which one could use as the limit of convergence.

Figure 10 shows how the GP model's global minimum prediction changes its location $\hat{x}$ and value $\mu(\hat{x})$ as more data is acquired. As a sign of convergence, both quantities decrease rapidly. While the decrease is not completely monotonic, it is consistent so that after crossing our convergence limit of 10^{-4} (dashed line) at 8 data points, it is never crossed back up again. Thus we would conclude, the global minimum was converged within our limit after 8 function evaluations. Relating to the GP model snapshots in Figures 8 and 9 the only visible changes in the minimum prediction (red vertical line) and its value on the y-axis occur when the third and fourth data point are acquired in Figures 8b and 8c. This is clearly reflected as the largest values in Figure 10.

This example illustrated how a BOSS search works. The sampling of the search space using an acquisition function was showcased in a concrete way, and the convergence of the minimum prediction was reflected between raw numbers and GP model snapshots. Most importantly I show in this example how easily the BOSS method is able to optimize a simple 1D function, leading the way to my results in more complicated optimizations in section 4.

3.3 BOSS acceleration

In this section I will discuss additional modifications to the BOSS method. They range from details of the Gaussian process (GP) to exploiting symmetries of the atomistic systems and are thus not so well connected to each other. However, the common factor for the subjects in this section is, that they aren't necessary parts of the method but can accelerate it.

3.3.1 Including gradients to GP model

The forces on atoms are often obtained from total energy simulations with little extra computational burden. By definition forces are negative gradients of energy, which are by default calculated in cartesian coordinates but can be projected to the used simulation variables $\mathbf{x}$. For example summing the inverses of the energy gradients' x-components of each atom in a molecule corresponds to the total force to translate the molecule in x-direction. This force is then gradient information, which can be added to the observations $(\mathbf{x}, \mathbf{y}) \rightarrow (\mathbf{x}, \mathbf{y}, \partial_{\mathbf{x}}\mathbf{y})$, and it can be utilized to improve the GP fit. The theory for this was explained in section 2.3.2.

The inclusion of gradient information improves the GP model fit based on increased knowledge about the direction and steepness of the slopes in the objective function. For example in 1D, two nearby data points with their gradients pointed in different directions indicate an extremum between them whereas without the gradients one cannot make a similar conclusion (see Figure 11).

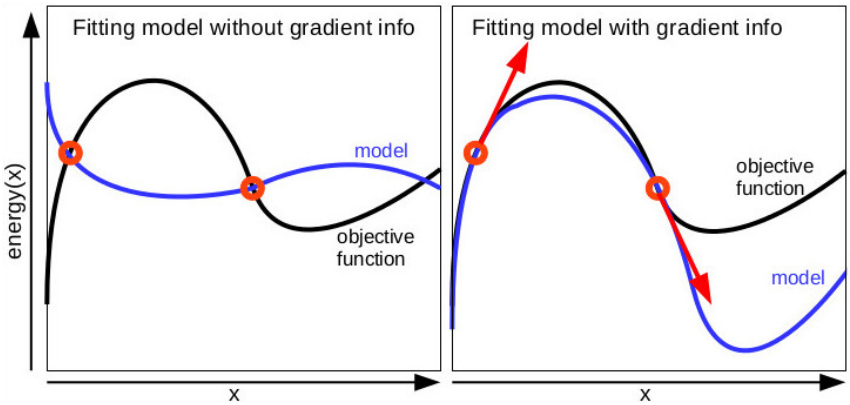


Figure 11: Illustration of how gradient information helps GP model interpolate objective function better with fewer acquisitions. Adjacent sampled points (red circles) with a gradient pointing first up and then down imply a maximum between them.

The gradient information also allows the GP to fit the length scale of the objective function more accurately with fewer data points. In higher dimensions the advantage

should grow due to the sampled points being further away from each other on average. The idea of gradient based improved GP fit in 1D is illustrated in Figure 11.

3.3.2 Mixing kernels

In the theory part of GPs in section 2.3.2, only a non-periodic kernel 10 and a periodic kernel 11 were introduced. However, there exist many applications in which some of the variables are periodic, while others are not. Perhaps the most important case like this in atomistic structure search is surface adsorption. The translation coordinates X and Y in the plane of the surface have periodic boundary conditions, but the Z coordinate doesn't. To apply periodicity (or any other kernel specific property) for selected variables only, kernels can be mixed. This is done by defining the kernels separately for periodic and non-periodic variables and then multiplying the kernels together. The kernel to be used for the GP model would be simply $\mathbf{K} = \mathbf{K}_{RBF} * \mathbf{K}_{STDP}$, where $\mathbf{K}_{RBF}$ (Eq. 10) acts on non-periodic variables and $\mathbf{K}_{STDP}$ (Eq. 11) on periodic variables. Using kernel mixing in this way, the X, Y and Z coordinates in surface adsorption problems can be optimized simultaneously with the correct boundary conditions. Thus excessive sampling of boundary areas (typical for non-periodic variables) can be avoided in case of X and Y, which makes BOSS more efficient.

Some systems have symmetries which cannot be taken into account with boundary conditions, but can nevertheless help make BOSS more efficient. For example mirror symmetries, rotational symmetries and combinations of those are very common in atomistic systems. To include the extra information to the GP model, the best option would be to use use a kernel, which takes into account the symmetry. If there is no such kernel, what one can always do, is to append multiple data points per evaluation. If for example BOSS queries $f(1)$ and we know that there is a mirror symmetry $f(x) = f(10 - x), \quad x \in [0, 5]$, we can return both $(1, f(1))$ and $(9, f(9) = f(1))$ to BOSS. Thus in this case the number of data points can be doubled compared to the number of function evaluations, which allows for the GP to be fitted much more efficiently.

3.3.3 Very high energy areas in search space

While BOSS can query the objective function value at any $\mathbf{x}$ within the domain χ we defined, there may be areas in that space which cause difficulties for the evaluation. In atomistic structure search such areas arise from unfavourable or unphysical structures. In principle the domain χ should be defined so that such structures would not belong to it, but in some cases that is not possible without piece-wise definition of space (not supported) or having to choose non-intuitive complex variables. This is the case in the

conformer search of many amino-acids, where the dihedral angles between functional groups are a natural choice for the degrees of freedom, but their rotations may cause atoms to clash (get too close to each other). Figure 12 shows an example of a simulation variable in alanine dipeptide conformer search, which at some of its values causes atoms to be placed very densely and thus energies of many orders of magnitude higher than usual to be returned.

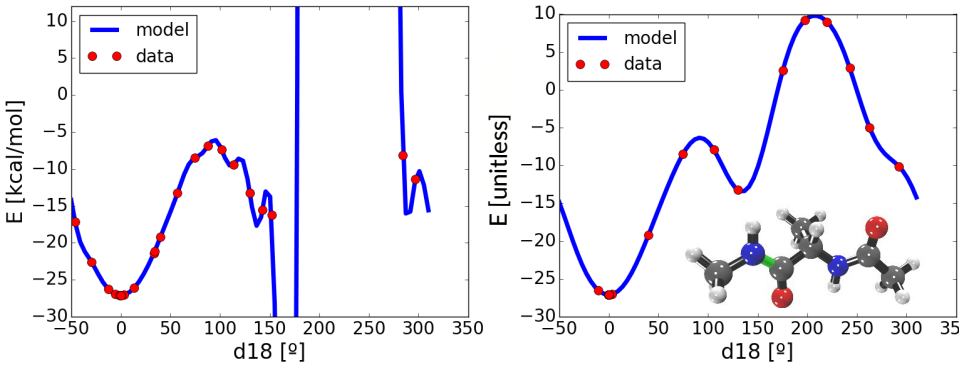


Figure 12: Truncating the energies returned by AMBER[1] forcefield for the rotation of one of the central dihedral angles (bond marked in green) of alanine dipeptide molecule. The left side graph shows the resulting GP prediction of a BOSS search with raw energy data. The right hand side graph shows the GP prediction of a BOSS search on energy data, which is truncated by taking a logarithm of values greater than 1.

The large peak in the energy values makes it extremely difficult for the GP to predict the length scale and variance (see hyperparameters in section 2.3.2) of the variable correctly. This makes the GP prediction very inaccurate before a large amount of data points are collected. By taking a logarithm of the high energies returned by the simulator, it is possible to produce a smooth truncated function, which is much easier to fit a GP on. In Figure 12 the truncation rule applied is:

$$\begin{aligned} E > 1 \quad &\rightarrow \quad E = \ln(E) + 1, \\ E \leq 1 \quad &\rightarrow \quad E = E. \end{aligned} \tag{18}$$

The logarithm truncation of course distorts the energies, but it doesn't change the position of minima, which one is looking for. It is important that the predictive accuracy is not changed, even though high energy regions with no minima or saddle points are distorted. Later on it is of course possible to do an inverse transformation back to energies returned by the simulator. Note that when taking a logarithm of the energies, we technically cannot talk about the same units anymore. Rather the quantity is then unitless E_1 with

a normalization of $E_1 = E/(\text{kcal/mol})$. However, below 1 (i.e. in the region of low energy configurations) the numeric values of calculated energy and unitless E_1 are equal.

Certain simulators cannot compute an energy at all for too dense configurations. In these cases one could in principle insert some artificial (high) energies instead of running a simulation, but this may lead to problems with the continuity of the function and its derivative. Another option is to modify the acquisition function so, that the problematic regions in the domain χ are simply never queried (e.g. by using cost-functions). In worst case one might have to divide the space in several regions, which are each optimized separately, and then compare the minima found for each subproblem.

3.4 Performance indicators

In the BOSS method the predictive GP model is iteratively refined as more data is collected (see Figure 1). Because the goal is to get an accurate model with as few objective function evaluations as possible, it is important to know when one can stop iterating. Thus in this section I present performance indicators for the BOSS method, that have been developed along with doing research with BOSS and developing and testing the code.

In one and two dimensions, the entire GP prediction can be easily illustrated by plotting it, and therefore one can in practice see when it stops changing. In more dimensions, however, one must rely on tracking carefully defined criteria. The basic principle is to stop iterating, when some measure (or several measures) of convergence reaches a predefined tolerance threshold. There exist several measures of convergence that one can track during the iterations. Which one to use, depends on what does one require of the model. In some applications it is enough to find just the global minimum of the objective function. This could be the case, if one knows beforehand (based on e.g. physical intuition or previous literature on the studied system) that there should only be one significant minimum, which is almost never guaranteed. Conversely, if one wants to conduct for instance a conformer search, the convergence of all local minima becomes the important factor to track. If, however, the interest lies in reaction paths (minimum energy paths), one must require the entire model to converge to the objective function before it is safe to stop iterating. The efficiency of BOSS method is intuitively defined as the number of objective function evaluations needed to reach chosen type of convergence.

Before I can describe the different measures of convergence, it is necessary to discuss and define the available quantities one can calculate on each iteration. These are all quantities calculated from the GP model and the data ensemble, which both get updated on every iteration. The model global minimum prediction $\hat{\mathbf{x}} = \text{argmin}(\mu(\mathbf{x}))$ is perhaps the most important quantity as it will tell us the best found structure of the system.

To find it one has to minimize the GP model mean $\mu(\mathbf{x})$, which may be cumbersome if the dimension of $\mathbf{x}$ is high. Related quantities which are acquired on the side are the predicted value $\mu(\hat{\mathbf{x}})$ and uncertainty $\nu(\hat{\mathbf{x}})$ at the global minimum. If one is willing to pay the computational price and make an extra evaluation of the objective function, the difference between model value and objective function value at predicted global minimum $f(\hat{\mathbf{x}}) - \mu(\hat{\mathbf{x}})$ can provide a very reliable measure of the correctness of the model (at that important location). The easiest quantity to follow is the lowest so far sampled energy y_{best} and its location $\mathbf{x}_{best}$ in the data ensemble. Because some data acquisitions are almost always taken near the global minimum prediction (if any exploitation is included in the acquisition function), the lowest sampled data point should be near $\hat{\mathbf{x}}$. For tracking the evolution of other areas than just the global minimum, a simple solution is to calculate the root mean square difference between the current and the previous iteration's GP mean: $\sqrt{\int(\mu_i(\mathbf{x}) - \mu_{i-1}(\mathbf{x}))^2 d\mathbf{x}}$. This is a full measure of how the much the model is changing from iteration to another, but its accuracy is limited by how accurately the N-dimensional integral is computed (with e.g. Monte Carlo integration).

The definition of convergence is that the target quantity stops changing within a tolerance. Therefore the change of global minimum prediction location $\Delta\hat{\mathbf{x}} = \hat{\mathbf{x}}_i - \hat{\mathbf{x}}_{i-1}$ and value $\Delta\mu(\hat{\mathbf{x}}) = \mu(\hat{\mathbf{x}}_i) - \mu(\hat{\mathbf{x}}_{i-1})$ dropping below a certain tolerance indicates that the global minimum is found up to some accuracy. However it is very common that in the early iterations, the model predicts the global minimum to be in one place but then suddenly changes the prediction someplace else, as more data has been acquired. The changes $\Delta\hat{\mathbf{x}}$ and $\Delta\mu(\hat{\mathbf{x}})$ can therefore be very small initially then fluctuate and settle below the tolerance again. For this reason the criterion for convergence using these quantities should also include the information that they have been below the tolerance for many enough iterations. If the objective function happens to have many local minima that are comparable in function value, it could be very slow to get $\Delta\hat{\mathbf{x}}$ to converge, because the model keeps changing its prediction about which one is the global minimum. In this case $\Delta\mu(\hat{\mathbf{x}})$ still remains a valid measure of convergence. As for the convergence of the model in other areas, one can track the root mean difference between consecutive models getting lower and the number of local minima to stop changing.

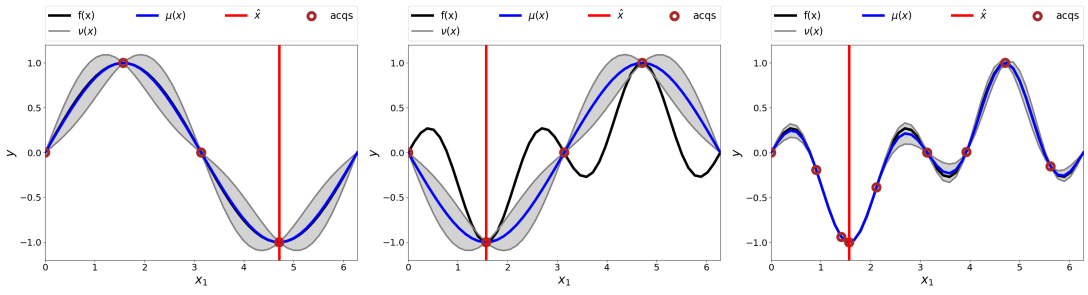
The presented quantities to track as criteria for convergence were:

- location $\hat{\mathbf{x}}$, value $\mu\hat{\mathbf{x}}$ and uncertainty $\nu\hat{\mathbf{x}}$ of the global minimum prediction
- true function value difference from model value at global minimum prediction $f(\hat{\mathbf{x}}) - \mu(\hat{\mathbf{x}})$
- best (lowest energy) acquired data point $y_{best} = f(x_{best})$

- model overall difference to previous model $\sqrt{\int(\mu_i(\mathbf{x}) - \mu_{i-1}(\mathbf{x}))^2 d\mathbf{x}}$

Now we can come back to efficiency. The efficiency of BOSS method is defined as the number of objective function evaluations until convergence. However, because the method has stochastic elements, the efficiency of a boss structure search is not the same every time it is run. It is especially the random way of optimizing the GP hyperparameters (see Section 2.3), which may fork the BOSS search on slightly different paths and result in convergence achieved at a different iteration. Due to this, the efficiency needs to be treated as a statistical variable. To determine the efficiency given an objective function and some convergence criteria, repetition of the search is required. Then the efficiency can be claimed to be the average number of iterations with standard deviation as the confidence interval (assuming it is normally distributed). Note that this type of efficiency determination makes sense only when studying and benchmarking the method itself – not when using it to solve novel materials science problems (as in section 4).

The results in this work (see Section 4) contain efficiency studies carried out with lightweight objective functions, as they are important in predicting how well the method can perform in larger accurate structural optimization problems. Studying how the optimization efficiency scales as a function of problem dimension can indicate, how feasible it would be to include more building blocks in the optimization. If the number of objective function evaluations until convergence scales well with the number of simulation variables, it indicates BOSS is useful for a wide range of atomistic structure search problems using *ab initio* simulators.



(a) $\sin(x)$ 4 pts (b) $\sin(x)\cos(2x)$ 4 pts (c) $\sin(x)\cos(2x)$ 9 pts

Figure 13: GP fits to a simple sine function $f(x) = \sin(x)$ and a more complex sine times cosine $f(x) = \sin(x)\cos(2x)$. The entire sine can be fitted accurately with just four data points whereas the latter one with three minima requires nine points for accurate prediction. The scaling of BOSS efficiency depends strongly on how complex functions are combined into a multidimensional problem.

It is important to understand, that the efficiency of BOSS structure search depends in

addition to objective function dimension also very much on its complexity. The two major components identified to affect function complexity (from optimization point of view) are the number of minima and maxima and the amount of correlation between variables in multidimensional functions. Even though the fitted GP model can adapt to a complicated function by optimizing the kernel hyperparameters (see Section 2.3), more data is still needed for accurate modelling compared to a more simple function. Figure 13 shows an examples of such a situation.

In more than one dimension, the correlations of the variables can also make a big difference. An objective function with completely uncorrelated variables (such as adsorption of atoms far away from each other) can be expected to be fairly easy to optimize, as it effectively reduces to several separate 1D problems. Then again a highly correlated function (such as adsorption of atoms into a cluster on the surface) is more difficult to optimize, as the change in one variable can change the energy in completely different ways depending on the values of the other variables. The scaling of the efficiency in case of different kinds of functions is studied in detail in the Section 4.

4 Results and discussion

4.1 Scaling study using simple analytic functions

4.1.1 Introduction

When exploring the possibilities and limits of a scientific method, it is important to find out how its efficiency scales when the problem size increases. Often it is possible to simulate problems with consistently increasing size, and from the results try to figure out a pattern for how long it takes to solve a problem of an arbitrary size. This way it is possible to estimate, how much time ans resources it would take to tackle a certain problem using the method in question.

For the BOSS method the size of the problem equals the dimension of the function that we are trying to optimize. The efficiency on the other hand corresponds to the number of function evaluations needed in the optimization. From that number it is straight forward to calculate the time taken, if a single evaluation's – i.e. one static atomistic simulation's – duration is known. Therefore using BOSS to optimize functions with increasing dimensionality, will reveal how the method scales. However, the efficiency is affected also by the complexity of the target function. Thus one should study the scaling using such trial functions that the level of complexity would be maintained while dimension is systematically increased. This way one will obtain a meaningful the scaling

curve that could be extrapolated to make predictions.

In this section I benchmarked the scaling of the efficiency of BOSS method using simple analytic functions. The level of complexity was made low by choosing functions with only a single minimum and relatively homogeneous length scale – i.e. the simplest possible cases of optimization. This provides a baseline for the applicability of BOSS to more complex optimizations. I compare functions of different types, boundary conditions and correlations between their variables. While artificial, the tested analytic functions were not dramatically easier to optimize than some potential energy landscapes in atomistic systems. The fast evaluation times however, make it possible to do systematic statistical testing of the scaling of BOSS efficiency.

There were five chosen test cases, which are from here on referred to as *suc*, *sucn*, *sc*, *huc* and *hc*. In this notation *s* refers to the sine function, *h* to the harmonic well function, *c* and *uc* to correlated and uncorrelated respectively, and *n* to non-periodic. Each of the five cases represents a set of functions, which have the same level of complexity but an easily tunable dimensionality. This way I can easily perform a scaling study and compare the effects of periodicity, correlation and function complexity on BOSS scaling.

The functional forms of the chosen function classes were the following:

$$f_{suc}(\mathbf{x};\mathbf{m}) = \sum_{i=1}^{dim} [\sin(2\pi(x_i - m_i + 0.75)) + 2.1] \tag{19}$$

$$f_{sucn}(\mathbf{x};\mathbf{m}) = \sum_{i=1}^{dim} [\sin(2\pi(x_i - m_i + 0.75)) + 2.1] \tag{20}$$

$$f_{sc}(\mathbf{x};\mathbf{m}) = \prod_{i=1}^{dim} [\sin(2\pi(x_i - m_i + 0.75)) + 2.1] \tag{21}$$

$$f_{huc}(\mathbf{x};\mathbf{m}) = \sum_{i=1}^{dim} [2(x_i - m_i) + 1.1] \tag{22}$$

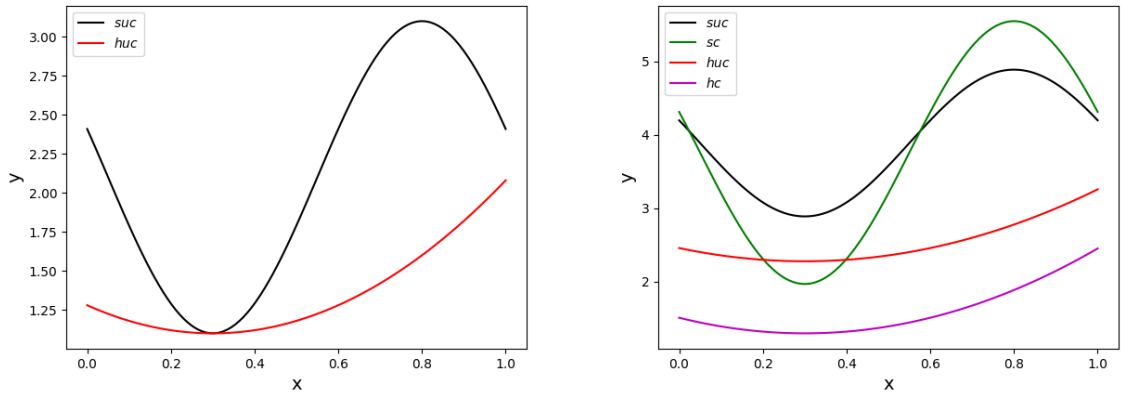
$$f_{hc}(\mathbf{x};\mathbf{m}) = \prod_{i=1}^{dim} [2(x_i - m_i) + 1.1]\,. \tag{23}$$

The first test case *suc* (Eq. 19) is simply a sum of sine functions added with a constant that makes $f_{suc}(\mathbf{x};\mathbf{m})$ be always positive. While *sucn*-case (Eq. 20) is defined as the same set of functions as *suc*, it is separated into its own test case by modelling it with a non-periodic kernel. While this would normally not make sense, it will in this case help us see the difference in efficiency scaling when including periodicity information and not including it. The other functions are modelled in normal manner with a kernel that matches their periodicity or non-periodicity.

sc (Eq. 21) also defines functions combining sines, but now as a product instead of

a sum. This makes the components of the variable array $\mathbf{x}$ be correlated – i.e. effect of changing the value of one component depends on the values of the other components: $f(\mathbf{x} = [a, c]) - f(\mathbf{x} = [a + b, c]) \neq f(\mathbf{x} = [a, d]) - f(\mathbf{x} = [a + b, d])$. Similarly as for the sines, an uncorrelated summed version and correlated multiplied version of the harmonic well function is the base of the function sets in Eq. 22 and 23. As these function have no periodicity, the test cases *huc* and *hc* (together with *sucn*) are modelled with a non-periodic kernel.

For each of the functions in Equations 19-23 the variable x_i can have any positive dimension and its domain is $\forall i, \quad x_i \in [0, 1]$. The parameters $\forall i, \quad m_i \in [0, 1]$ conveniently define the location of the (only) minimum in the functions. The parameters are present to make it easy to do unbiased statistics for the BOSS efficiency of finding the global minimum of these functions. As the parameter array $\mathbf{m}$ is randomly drawn for each BOSS search, the statistics on efficiency are not affected with e.g. the locations of the initial evaluation points (first two Sobol sequence points).



(a) f_{suc} and f_{huc} in 1D

(b) 1D profiles of f_{suc}, f_{sc}, f_{huc} and f_{hc} in 2D

Figure 14: Illustrations of the used functions. In (a) the functions are shown in 1D with their minimum (parameter m) at 0.3. Only two functions are shown since the rest reduce to the same functions in 1D. (b) shows 1D cross-sections of all the functions in 2D sliced with the second variable component at 0.5 while the minimum is at $m = [0.3, 0.7]$.

The 1D and 2D versions of each function class are illustrated in Figure 14. 14a shows them in 1D, in which case all sine-based classes (*suc*, *sucn* and *sc*) reduce to the same sine curve, and *huc* and *hc* reduce to the same harmonic well. In 2D in Figure 14b the 1D profiles of four different functions are to be seen, while *sucn* is not shown separately as it always equals *suc*. In the 1D slices, the uncorrelated cases *suc* and *huc* are simply the same curves as shown in plain 1D but shifted by a constant. On the other hand,

the 1D-sliced curves of the correlated functions *sc* and *hc* are scaled by the contribution of the second variable (a factor of 2 roughly in this case), so they look stretched out in y-direction. Nevertheless the minimum of all the functions is at the same location determined by the parameter $\mathbf{m} = [0.3, 0.7]$.

The lowest possible function value is $1.1 * dim$ for the uncorrelated function classes *suc*, *sucn* and *huc*, and 1.1^{dim} for the correlated classes *sc* and *hc*. This minimum is reached only when $\mathbf{x} = \mathbf{m}$. Similarly the maximum function values are $3.1 * dim$ and 3.1^{dim} for uncorrelated and correlated respectively. The variables in the uncorrelated functions do not affect each others contribution to function value (because their contributions are simply summed), whereas the variables in correlated functions do (contributions multiplied).

Now that the used functions are carefully introduced, I come back to the scaling study. Each of the five test cases will be optimized with BOSS starting from 1D and working up towards higher dimensions. Each optimization is continued until the global minimum prediction converges to the known correct answer ($f(\mathbf{x} = \mathbf{m}) = 1.1 * dim$ or $f(\mathbf{x} = \mathbf{m}) = 1.1^{dim}$) within a tolerance of 0.01 for all $|d\mathbf{x}_i|$ and $|d\mu(\mathbf{x})|$. This is repeated 10 times for each function, while drawing a different random parameter $\mathbf{m}$ every time. The statistics of function evaluations until convergence are collected for each case for each dimension until the optimization times get too long.

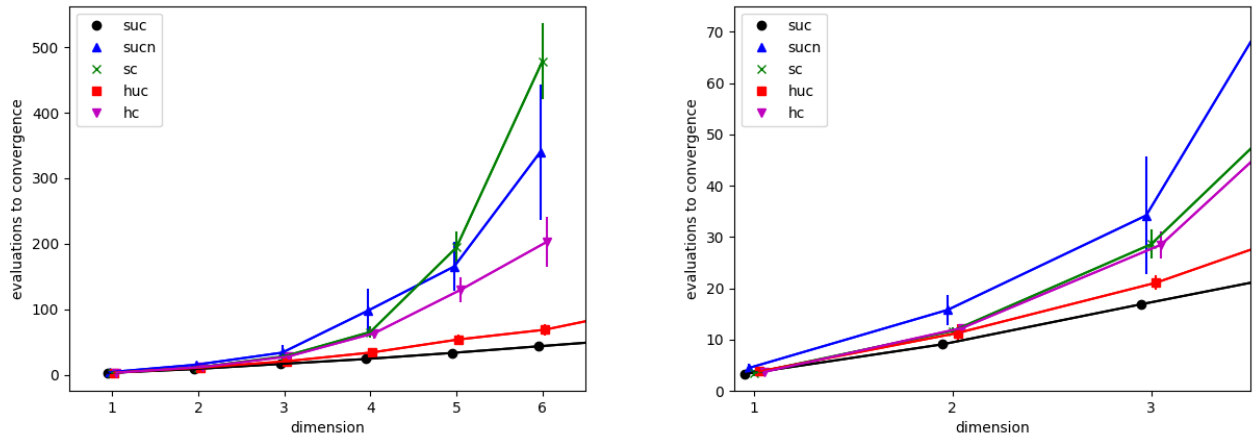
To be able to draw predictive conclusions of the resulting scaling curves, I additionally try to fit a linear ($ax + b$), parabolic ($ax^2 + bx + c$) and exponential (be^{ax}) model on the data. In these formulas x marks the dimension. The chosen models are a few general trials, which I expect to be able to fit the scaling curves if they rise consistently with number of dimensions. If a good fit is made, it will already predict the limiting behavior (e.g. $\mathcal{O}(x^2)$) of how BOSS scales in case of that function class. Moreover, the main coefficients (denoted a in each model above) in the fits can quantify the scaling difference between similarly shaped curves.

4.1.2 Results

Now I present the results of the scaling study with analytic function classes, and the associated trial fits on the scaling data. The BOSS efficiency as function evaluations until global minimum convergence, is treated as a Gaussian statistical variable for each dimension of each test case. The averages and standard deviations of the five cases named *suc*, *sucn sc*, *huc* and *hc* up to 6D are presented in Table 1 and plotted in Figure 15.

The results show that the ordering of the test cases' scaling curves is *suc-huc-hc-sucn-sc* in steepening order. The curves maintain their ordering from early on except for *sc*, which behaves up to 4D very similarly as *hc* and then rises rapidly to become the steepest

scaling curve by 6D. In 4D the global minimum of the easiest case *suc* (periodic and uncorrelated) was converged using 24.5 function evaluations on average. The other cases required several times as many evaluations. In higher dimensions the differences grow.



(a) Full result

(b) Zoom into dim 1-3

Figure 15: Function evaluations until convergence of global minimum prediction for tested functions. Subfigure (a) shows the full result extending up to 6D, while (b) zooms in to show the details in lower dimensions. Each data point is the average, and the error bar one standard deviation, of the same 10 times repeated BOSS search. A slight offset in the x-axis value is introduced to the lines as a visual aid to prevent the error bars from overlapping.

The error bars on the scaling data points show the standard deviations of convergence from the 10 BOSS runs making up each point. It can be seen that the absolute errors consistently increase in higher dimensions, but differently for the test cases. **sucn** has in 6D a larger error than there is for **sc** although the latter took roughly a hundred more evaluations to converge on average.

The scaling data stops at 6D, because the time of a single BOSS search started to become too long. While for the easiest case *suc* the 7D with roughly a hundred iterations to convergence on average would have been well feasible, in the worst case *sc* the ≈ 500 iteration searches each took already about 50 hours. The function evaluations are done in split seconds for the analytic functions, but the fitting of the multidimensional GP model, optimizing the hyperparameter values and minimizing the acquisition function started to take up a lot of time for many dimensions and many data points. In hindsight, significant amounts of time could have been saved, if I had used fewer and less accurate gradient-decent walkers to minimize the acquisition function on each iteration.

case	**suc**	**sucn**	**sc**	**huc**	**hc**
pbc	yes	no	yes	no	no
corr.	no	no	yes	no	yes
1D	3.3 ± 0.9	4.5 ± 0.8	3.5 ± 0.7	3.8 ± 0.4	3.7 ± 0.6
2D	9.1 ± 0.9	15.8 ± 3.0	11.7 ± 0.8	11.3 ± 1.7	12.3 ± 0.5
3D	16.9 ± 0.5	34.2 ± 11.5	28.6 ± 2.9	21.1 ± 1.4	28.5 ± 2.6
4D	24.5 ± 1.5	98.4 ± 33.3	65.7 ± 8.7	34.6 ± 4.3	64.1 ± 8.7
5D	33.8 ± 0.9	165.5 ± 37.7	194.5 ± 24.8	69.1 ± 8.1	130.0 ± 18.8
6D	43.8 ± 0.9	340.0 ± 103.1	478.5 ± 57.5	96.2 ± 11.3	203.0 ± 38.3

Table 1: Average function evaluations until convergence of global minimum prediction for the five test cases. The error ranges represent one standard deviation interval to both directions. In the labels pbc is abbreviation for periodic boundary conditions, and corr. stands for correlated (variables).

In Table 2 I present the errors and leading coefficients a of the linear, parabolic and exponential models, which were fitted on the scaling data.

	linear fit $ax + b$			parabolic fit $ax^2 + bx + c$			exponential fit be^{ax}		
case	R^2	RMS	coef. a	R^2	RMS	coef. a	R^2	RMS	coef. a
suc	0.9994	1.96	9.18	**1.0000**	**0.58**	**0.26**	0.9720	14.00	0.32
sucn	0.9729	47.08	62.59	**0.9977**	**13.75**	**18.05**	0.9893	29.57	0.85
sc	0.9561	86.43	84.59	0.9950	29.15	32.62	**0.9997**	**6.96**	**0.97**
huc	0.9950	5.78	15.21	**0.9997**	**1.48**	**1.61**	0.9770	12.43	0.51
hc	0.9822	23.39	39.58	**0.9997**	**2.98**	**9.30**	0.9780	25.96	0.80

Table 2: R^2-values, RMS errors and leading coefficients of linear, parabolic and exponential fits on the five scaling curves. The best fits (smallest RMS) for each case are highlighted in bold.

For a perfect fit the R^2-value should be one and the RMS error should be zero. Thus in Table 2 I have highlighted as best fits for each test case, those fits that have the highest R^2 and lowest RMS combination. I only show the value of the coefficient a for each fit, because it is the most important coefficient in the models considering how the BOSS efficiency scales. The results show that the best matching fit would be the parabolic fit for all other test cases except *sc*, which is best fitted by the exponential model. The ordering of the cases fitted by a parabel is – according to the second order coefficient a – in steepening order *suc*, *huc*, *hc* and *sucn*.

4.1.3 Analysis and discussion

The monotonically and consistently increasing scaling curves and their errors indicate, that the chosen function classes managed to maintain the level of complexity rather well while the dimension increased. This was one of the goals of this scaling study, as otherwise it would have been difficult to draw any definite predictive conclusions about BOSS scaling. The fact that the curves maintain their ordering (apart from *sc*) and that it matches the ordering suggested by the fits and their coefficients, makes one able to confidently deduce there are fundamental differences between the scaling of the test cases.

The results show that *suc* scales best – i.e. number of function evaluation needed for BOSS to optimize it increases the least as problem dimension increases. This was to be expected since *suc* is the only function with both periodic boundary conditions and no correlation between variables. Its scaling ($\approx \frac{1}{4}dim^2$) directly sets the baseline for the others, especially for *sucn* and *sc* which introduce non-periodicity and variable correlations (respectively) as additional complications on top of *suc*. The use of non-periodic GP model in case *sucn* makes the scaling worse by roughly $18 * dim^2$ but it is still parabolic. The test case *sc* scales even worse as its scaling turned out exponential ($\approx e^{dim}$). Nevertheless the errors of *sc* scaling data points are much smaller than for *sucn*. Thus in case of the function classes with sine functions, variable correlations made the scaling worse but decreased robustness less than lack of periodicity.

The cases with the harmonic function – *huc* and *hc* – show scaling somewhere between the sine cases. *huc* scales as $1.6 * dim^2$, which is much better than its non-periodic but uncorrelated counterpart *sucn*. *huc* scaling is also much more robust. I think the reason for this lays in the extremely simple shape of the harmonic well function, which in any dimension monotonically decreases from the bounds towards the single minimum somewhere in the middle. Lack of periodic boundary conditions is known from previous work to lead to BOSS acquiring many data points at the boundary areas, because the uncertainty is large near the unknown region outside the bounds (GP is defined in all the space even if we constrict the search to a certain interval). As the hypersurface of the boundary areas increases as one goes to higher dimensional functions, the problem of excessively sampling the (often uninteresting) boundary areas becomes worse and makes the scaling curve steep. For *huc* this effect is much reduced by the fact that function value is always increasing towards the bounds. On the other hand, for the sine based functions in test case *sucn*, the function value is decreasing towards the bounds in roughly half of the times, making BOSS direct much of the sampling there even though the actual minimum is always somewhere in the middle. This largely explains the difference between the scaling of *sucn* and *huc*.

hc, which introduces variable correlations on top of *huc*, made the scaling worse but by much less than *sc* compared to *suc*. The robustness of *hc* and *sc* is roughly the same. As *suc* and *huc* scale relatively similarly but *sc* and *hc* so differently, must be due to either periodicity or the difference in the shapes of sine and harmonic well. As periodic boundary conditions are extra information to the GP model, they could hardly explain *sc* being more difficult to optimize. On the other hand, in sine function the gradient varies rapidly and changes sign, while in harmonic function the change of the gradient is slow an monotonic (2nd derivative is constant). I think this makes it more difficult for BOSS to get the length scale of the GP model fitted correctly to the data. Consequently wrong length scale makes the GP model inclined to over-fitting or under-fitting, and therefore cause acquisition function to direct data acquisitions to the wrong places. Here wrong means non-informative data points e.g. close by a previous data point far from the actual minimum. This in turn increases the number of function evaluations needed to optimize the target function. Thus I deduce that when variable correlations are added to play, the steepening of the scaling curve is greater for the sine based function classes over the harmonic well function because of the more complicated function shape. While both functions only have one minimum and are smooth, the difference in the complication of the functional shapes can be understood as a difference in the functions' first and second derivatives.

The order of the test cases' scaling curves – *suc*, *huc*, *hc*, *sucn*, *sc* in steepening order – is by the above reasoning explained in addition to the original divisions based on periodicity and correlation, also by the detailed shape of the underlying base functions: sine and harmonic well. With the obvious factor of function complexity – number of minima – set to constant one for all used functions, the functions derivative's simplicity and function behavior near boundaries (for non-pbc) were found to be important factors as well.

The most important outcome of this study is the finding and comparison of factors affecting the scaling, not so much the exact scaling of each of the test cases. It is true that the fitted models allow us to extrapolate somewhat and confidently predict that e.g. *huc* class function (and similar non-periodic single wells) in 10D could be optimized with roughly 200 function evaluations on average. However, six data points is still fairly small data and there is yet more value in knowing what kind of features of the optimized function affect the scaling. Based on this result one can expect the established order of the test cases' scaling to remain, even though the scaling would actually turn out exponential for all of them, when higher dimensions are taken into account.

4.1.4 Conclusions

This study compared the scaling of BOSS efficiency to optimize different analytic function classes as the dimension of the functions increased. The differences between the test cases included periodicity, variable correlations and two differently shaped base functions, which were combined to make up a higher dimensional function of the same complexity level per dimension. The scaling test reached 6D and showed that four of the least steep curves could be fitted accurately by the slope of a parabel, while only the steepest curve was better fitted by an exponential model.

Importantly, the scaling curves proved to be monotonic and consistent, and reveal clear differences between the test cases. Based on these I could interpret that while periodic boundary conditions are known to make BOSS optimization much more efficient, also the correlations between variables and the complication of a function's derivatives play a significant role in the scaling. For non-periodic functions also the behavior of the function near the boundaries matters. Knowing this helps us in planning how to choose the simulation variables, that we are optimizing using BOSS. It also allows one to estimate roughly how many function evaluations one might need before convergence, and thus knowing beforehand what kind of BOSS search is feasible and what is not.

From a BOSS user's point of view, the above mentioned properties affecting the scaling can be viewed as a list of properties to favor when choosing the simulation variables. If there is some periodic symmetry in the system, it is very beneficial to take the entire period as the search space and so make the corresponding variable periodic. Even making a naturally non-periodic variable periodic by introducing some artificial connection between the boundaries could be considered (provided the minimum is known not to be at the boundary). On the other hand if some simulation variable seems very complex (e.g. fast varying gradient) already in 1D, one should consider making the approximation of solving that variable separately from the others. Then the main optimization (including all the other variables) could be done more than once while fixing the value of the complicated variable to some of its lowest minima at a time. Thus savings in the number of function evaluations can be made, but one can still find global minimum energy configuration of the system with good confidence.

4.2 Conformer search of alanine dipeptide using gradient observations

4.2.1 Introduction

This section features results of a BOSS study on the conformer search problem of the alanine dipeptide amino acid molecule (Figure. 16) in gas phase. We apply BOSS to solve the global minimum energy conformer and then compare the efficiency and quality of the result to literature results. The goal of this is to demonstrate BOSS method's performance in a realistic problem, but one that is well known in literature so it can easily be compared against other methods.

As a simultaneous study, the scaling of BOSS efficiency is mapped along the way by taking the simulation variables into account one-by-one before performing the actual full search, whose efficiency can be compared to literature. Additionally, all of the BOSS searches are done both with gradient information (see 2.3.2) and without. So we can identify the efficiency gain of gradients both in the scaling curve and the final result's efficiency.

Apart from being a commonly well studied system, the alanine dipeptide molecule was chosen for this study, because it allows one to use a fast classical forcefield potential AMBER[1] for the energy evaluations, while still maintaining reasonably good physical accuracy. Even though *ab initio* methods would have been feasible to use for such a small molecule, the fast evaluations were now desirable in order to be able simultaneously study the optimization efficiency statistically by repeating the BOSS searches several times (as explained in section 3.4).

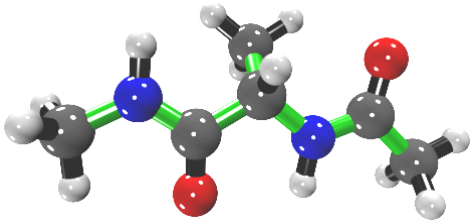


Figure 16: Alanine dipeptide molecule. The seven green bonds are the ones that are rotated to reach new conformers.

Alanine dipeptide is a well known molecule and its conformers have been mapped in many studies[20, 25, 16, 23]. It is an amino acid terminated with acetyl and methyl amide, and an important "building block" in proteins. The dihedral angles between the backbone bonds mostly determine the configurations of different stable conformers, which has been

found not only empirically but also using machine learning methods[8]. Additionally the potential and free energy landscapes of alanine dipeptide – which BOSS method models directly – have been studied[10, 23]. From the landscapes it is possible to extract in addition to the stable conformers also the energy barriers between them. In gas phase alanine dipeptide has two important conformers: β which has the backbone straight, and $C7_{eq}$ which features hydrogen bonding between oxygen and the hydrogen in the amide group (see Figure 18).

The chosen simulation variables were the torsional angles along the backbone of the molecule (see Figure 16) – also called dihedrals. They correspond to rotating parts of the molecule with respect to each other, with the backbone bonds as the rotation axes. This means that all the bond lengths and bond angles were fixed and the molecule reaches different conformers by the torsional rotations. `OpenBabel` [2] was used to switch between cartesian representation (`.xyz` file) and Gaussian z-matrix representation (`.gzmat` file) of the molecule, where the later one allows one to directly access the dihedral angles. A few of the possible dihedral angle definitions correspond to a motion of folding the molecule in a way that would definitely increase the potential energy (by putting atoms very near to each other), so those were discarded from the set of simulation variables by fixing their values. Varying the dihedral angles allows the molecule to reach all the relevant conformer configurations while still allowing us to ignore those degrees of freedom (bond lengths and angles) that are not likely to change significantly anyway, since they have a single simple equilibrium value (like e.g. the structure of the methyl groups). Notably the dihedral angles as simulation variables produce functions which are strongly correlated and have multiple minima, indicating a greater challenge for the BOSS method to optimize compared to single-well-like simple functions considered in section 4.1. The Gaussian z-matrix representation of alanine dipeptide molecule defines 19 dihedral angles, but removing those that are dependent on the others or fold the molecule in an impropable way, I selected only 7 (named d4, d8, d10, d12, d16, d18 and d20). All of them are naturally 360 degrees periodic with the exception of three (d4, d12 and d20), which are 120 degrees periodic due to the symmetry of the methyl groups. The latter three dihedrals correspond to a movement of rotating the methyl groups. The rest correspond to rotating two parts of the molecule with respect to each other, while keeping one of the backbone bonds as the center and axis of rotation. Whenever a lower dimensional BOSS search than 7D is considered in this study, the other dihedrals are being fixed to their default positions as shown in Figure 16 and Table 5.

The static potential energy calculations were done with the AMBER[1] package's classical forcefield simulator called `sander` that has experimentally fitted parameters which

work especially well for proteins and thus for our aminoacids. The calculations are really fast (≈ 0.1 s / static calculation) which is the main reason for this choice of code. In some configurations within the domain of the 7 simulation variables of alanine dipeptide, very high energy configurations can be reached as atoms come very close to each other. From a conformer search point of view, these are not at all interesting configurations but because they are within the domain of the BOSS search, they need to be included. A good example is the d8 angle of alanine dipeptide shown in original its form of the left side of Figure 12. It has such a large peak at around 220° that without zooming in, the other areas appear completely flat. There is, however, meaningful fine structure in the other areas and an obvious global minimum. This can be seen on the right side of the same figure where the energies returned by the simulations have been truncated with the rule shown in Equation 18. The rule was applied to all the energies returned from AMBER in this study. Due to the fast AMBER simulations, the gradient observations were calculated by making extra evaluations while using the finite difference method $\frac{\partial f(x)}{\partial x} \approx (f(x-\epsilon) - f(x+\epsilon))/2\epsilon$, even though we could also have calculated it analytically from the forces on atoms. We did just that for some of the dihedrals to try it out. While it wasn't too difficult a calculation to implement, it seemed unnecessary to spend much time on while doing this work.

4.2.2 Scaling with and without gradient information

Following the principles explained in section 3.4, the scaling of the efficiency of BOSS to find the global minimum was determined both with and without using gradient observations. All the combinations of 1,2,3,4,5,6 or 7 out of the seven possible simulation variables were searched separately by BOSS, and the number of acquisitions[3] until global minimum convergence within 0.1 kcal/mol was averaged over searches of same dimension. For the seven-dimensional search there is only one combination of variables to choose, so that search was repeated ten times to provide better statistics in that dimension as well. The resulting scaling graph is shown in Figure 17.

It can be seen that at low dimensions, the accelerating effect of gradient observations on the global minimum convergence is not very significant (< 10 evaluations), but especially after 3D the difference becomes notable and grows. In 5D, the BOSS runs with gradients converges already with half the points needed for the non-gradient run on average. The slope of both curves is steeper than linear – possibly a higher order polynomial or exponential curvature. At 6D there is a slight elevation in the scaling curve for the gradient-including model that does not seem to follow the trend of the other data points. I believe the poor statistics can be blamed for this as there exist only 7 different 6D

[3]ignoring extra evaluations for calculating the gradients

variable combinations over which the average is taken, while for other dimensions there are tens of different combinations. The elevation is also well within the calculated error of one standard deviation.

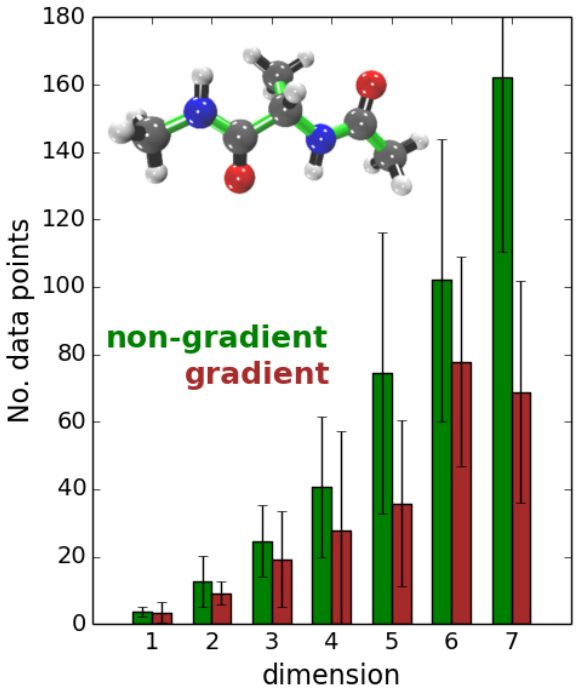
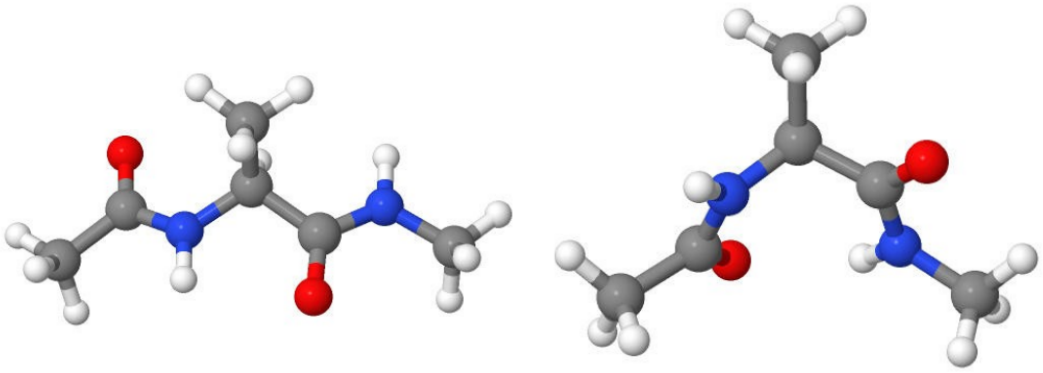


Figure 17: Average number of data points until global minimum convergence as a function of dimension for non-gradient model and gradient-including model.

4.2.3 Conformers

Apart from the scaling, the conformer search is realized by extracting the local minima out of the final GP model in the full 7D BOSS search. Each minimum corresponds to a stable conformer and, if low in energy, a conformer that might appear frequently in nature. The two lowest energy conformers found are shown in Figure 18 and five lowest listed with detailed values in the appendix in Table 5. The molecular configurations of the two lowest match to the previously presented β with straight backbone and $C7_{eq}$ with internal hydrogen bonding. While the ordering of the energies of these two conformers may differ in results done using other forcefields, they are still distinctively lower in energy than any other conformers[23]. In our result, the 3.-5. lowest local minima are essentially duplicates of the second lowest $C7_{eq}$ as they vary mostly only by the positions of the methyl groups, which matter little to energy. The next lowest minimum after that (6th lowest) is about 5kcal/mol higher in energy. An exact match to literature results wasn't to be expected, since we used a force field different from others. In our result, the energy

difference between β and $C7_{eq}$ is 2.22kcal/mol in favor of β (global minimum). Strodel et al[23] only show the equivalent result for free energies at 289K using a different forcefield, and it states $C7_{eq}$ is a 0.15kcal/mol lower minimum than β.



(a) β - -28.59 kcal/mol (b) $C7_{eq}$ - -26.37 kcal/mol

Figure 18: The two lowest found conformers of alanine dipeptide in gas phase. They match to literature results.

The 7D BOSS search with gradients that found the conformers in Figure 18 and formed the PES model in Figure 19 made 374 data acquisitions. Because we didn't implement the gradient calculation for all the simulation variables but did it using the finite difference method, each data acquisition here made $1+2*7 = 15$ static energy calculations. However, we compare now the number 374 to other methods, as the actual number $15*374 = 5610$ could have been cut down to 374 by implementing the analytic conversions from atomic forces to dihedral gradients.

To put BOSS method's performance into perspective I compare to other methods' efficiency[4] in solving the same problem. Strodel and Wales (2008)[23] applied basinhopping algorithm along with extended harmonic superposition approach and separately replica exchange molecular dynamics (REMD) to calculate the conformers and the free energy surface (FES) of alanine dipeptide. The FES was spanned by the central dihedrals $d8$ and $d16$ (also named ϕ and ψ). They produced FES maps of similar resolution to our PES maps19, and additionally they claim their maps to be of similar quality to maps resulting from accelerated MD, metadynamics, umbrella sampling[16] and the single-sweep method. Producing the FES with this resolution implies also finding at least the two most stable conformers (see Figure 18). They accomplished this with basinhopping using 2000 iterations, which means 2000 energy minimizations consisting of several energy evaluations

[4]Note that here efficiency is considered in terms of the number of energy calculations regardless of the fact that in case of forcefields the calculations are very fast.

each. Thus energy must have been evaluated some 6000-20'000 times. With REMD they constructed the FES using 50 ns of simulation time. While they do not mention the used time step, it is safe to assume it was at least smaller than 500 fs. This indicates a lower limit of 100'000 energy evaluations. As BOSS was able to accomplish the same goal using 374 energy evaluations (neglecting here the gradient calculation evaluations), it means the effort reduced below 1% in terms of number of static simulations.

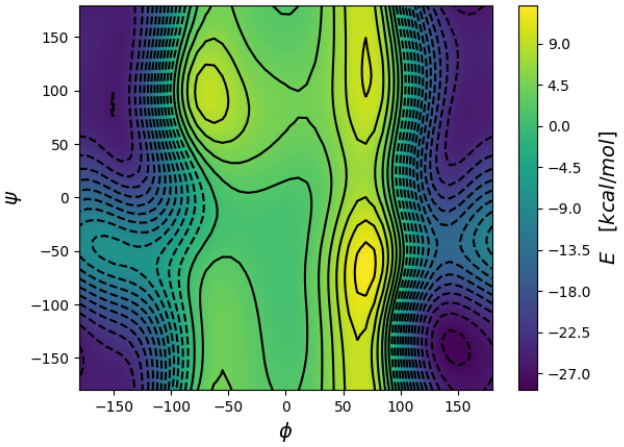


Figure 19: 2D PES spanned by the central dihedrals $d8$ (ϕ) and $d16$ (ψ) cut from the 7D map from BOSS search.

4.2.4 Conclusions

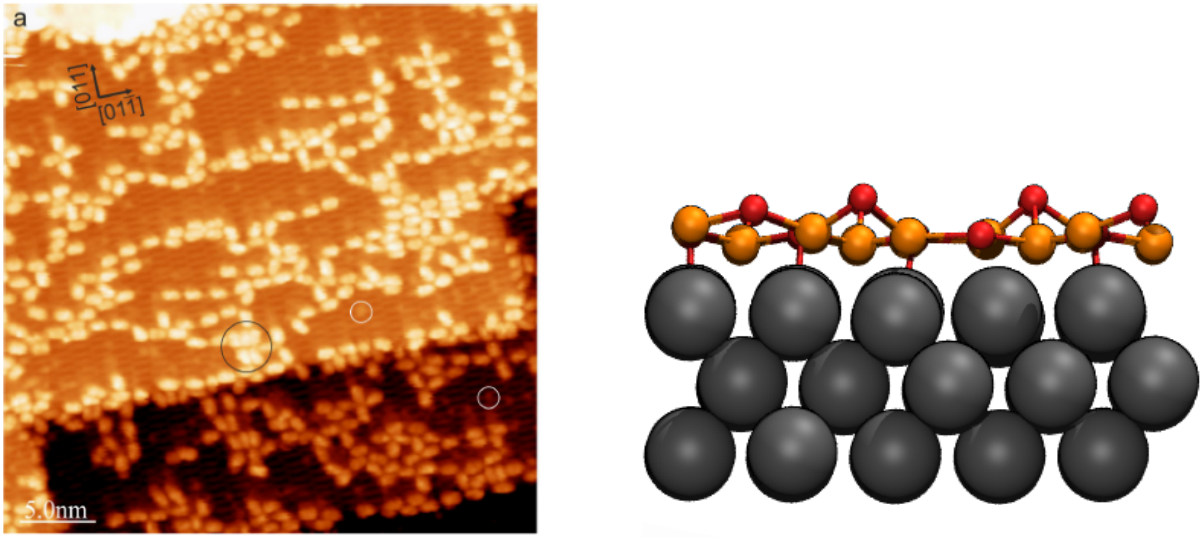
In conclusion, this study was successful because physically accurate results were obtained with a greatly reduced effort compared to other methods. Thereby BOSS comes out as a viable method to use in conformer search. Additionally, the benefit of including gradient information at observation points to BOSS was shown to more than half the number of simulations required. This encourages the use of gradients whenever atomic forces are available from the simulations.

4.3 Bifenyl dicarboxylic acid on cobalt oxide thin film

4.3.1 Introduction

In this study we applied BOSS to discover the energetically most favorable adsorption configurations of bifenyldicarboxylic acid (BDA) on CoO thin film on Ir substrate. The same system has been studied earlier[7] and DFT calculations have been used to try and find the adsorption configurations based on analysis of experimental data. The tentative

adsorption structures found before[3] featured a deprotonated BDA on a 6x5 supercell of 1BL film of CoO on 3 atomic layers of Ir(100).



(a) (b)

Figure 20: On the left (a) there is an STM image of the adsorption of BDA molecules on 1 BL CoO film on Ir substrate (from A.Schneider). On the right (b) there is a side view of the simulation cell. Iridium is shown in grey, copper in orange and oxygen in red.

This study started as an attempt to explain experimental images provided by A.Schneider (see Figure 20a). The idea was to try and find lower energy adsorption sites and configurations using BOSS method, than what he had found by making DFT simulations of relaxing the system from a few starting structures inspired by the images. For the simulations he and we used a simulation cell with three atomic layers of the Ir lattice, on top of which there is one layer of CoO (see Figure 20b) with alternating atom heights. In the top view of the system in Figure 24 the unit cell on the surface can be seen. It is a tilted rectangle with respect to the underlying rows of iridium. The simulation cell contains three repetitions the unit cell. The BDA molecule (Figure 21) that is adsorbing, consists of two attached benzene rings with terminating oxygen pairs on the far side of each ring. This is the deprotonated version of the molecule, which is observed to adsorb. The initial expectation is that the oxygens at the ends of BDA are the reactive parts, determining how the molecule will anchor on the surface.

Static DFT simulations were used to calculate the energies corresponding to atomic configurations. The calculations were carried out at the PBE-PAW level employing the DFT+U approximation for cobalts and Grimme vdW-D3. The substrate was a 6x5 supercell of 1BL film of CoO on 3 atomic layers of Ir(100) and the molecule was a deprotonated

version of the BDA. The simulations were done using the VASP code on CSC taito supercomputer using 128 cores. Every BOSS iteration's time was dominated by these VASP simulations which took roughly 20 minutes including a separate short simulation for the molecule's energy in isolation to be able to calculate the adsorption energy.

Workflow of this study was to first study the BDA molecule in isolation to explore its internal degrees of freedom, and then place the molecule on the substrate to look for the adsorption sites. Both the molecule's internal and the relative simulation variables between the molecule and the substrate were chosen so, that we believe all relevant chemistry should be captured in the BOSS optimizations. Nevertheless, we took the minimum energy adsorption configurations from BOSS results and started structural relaxations from them. This allows also those atoms to relax which we had fixed in the BOSS searches. The relaxed configurations were then compared to those found in ealier studies.

4.3.2 Results on isolated BDA

First we studied the deprotonated BDA molecule in isolation in a large simulation cell ($25.0 \times 15.0 \times 19.4$ [Å]). The simulation variables (or degrees of freedom) to be optimized in BOSS were taken as the rotation between the benzene rings (noted C) and rotations of the terminating carboxyl groups (noted $O1$ and $O2$). See Figure 21 for an illustration of the rotations.

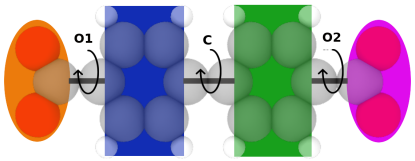
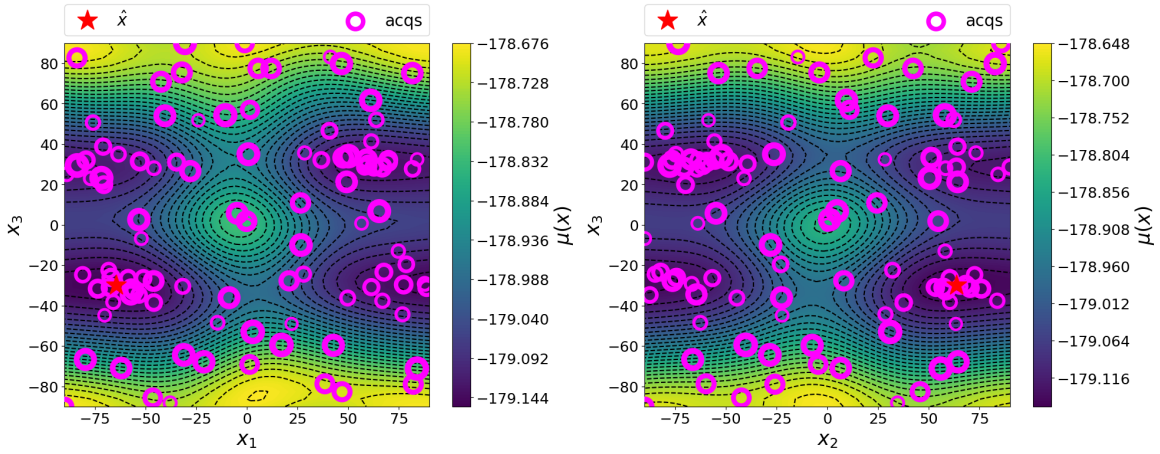


Figure 21: Rotations of the benzene groups and terminating oxygens of the bifenyldicarboxylic acid are taken as the simulation variables internal to the molecule. The variables named O1 and O2 rotate only the carboxyl groups, while the C-variable rotates one half of the molecule. Thus the O2-variable remains the angle between the carboxyl group and the nearest benzene ring.

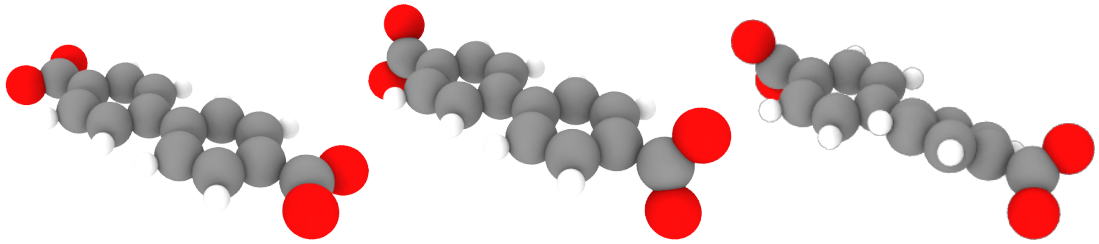
BOSS optimized all of the three simulation variables simultaneously and based on 102 static simulations constructed a surrogate model of the energy landscape (see Figure 22 for 2D slices of the produced 3D maps). It revealed that there exist 8 energetically identical minimum structures, as each of the variables has two mirror symmetrical minima around 0° (planar configuration). The variables proved to be nearly independent of each other. The minimum structures are all non-flat configurations of the molecule with twisting angle values $O1 \approx \pm 60°$, $O2 \approx \pm 60°$ and $C \approx \pm 30°$ with respect to flat configuration. Each of the three twists contributes roughly -0.2 eV to the molecule energy compared to the flat reference structure (see Figure 23). As a conclusion, the BDA molecule strongly favors

twisted configurations in isolation.



(a) O1-C (b) O2-C

Figure 22: O1-C and O2-C cross-sectional planes of the energy landscape produced by BOSS.



(a) Flat 0.0 eV (b) Carboxyls twisted -0.4 eV (c) Global minimum -0.6 eV

Figure 23: Three levels of lowering the energy of the BDA molecule. Note that for (b) there exists another energetically identical minimum with the carboxyl groups twisted in opposite directions to each other. Similarly for (c) there exist 7 others differing by the directions of the twists.

The flat molecule structure (Figure 23a) is referred to as just **flat**, while the twisted global minimum (Figure 23c) is referred to as **isogm** (for isolated global minimum).

4.3.3 Results on BDA on Ir-CoO

The next step was to place the BDA molecule on the Ir-CoO substrate in order to look for adsorption sites and configurations. To keep the problem dimension low and thereby acquire approximative results with a relatively small number of simulations, we fixed the shape of the BDA molecule and moved it rigidly on the substrate. The simulation

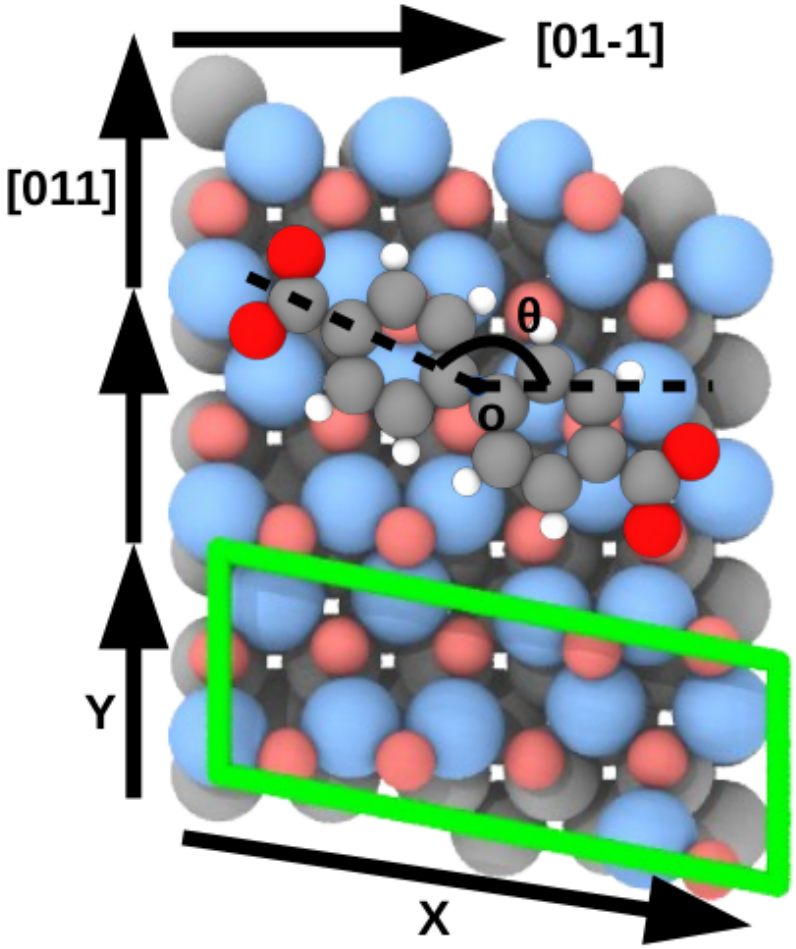


Figure 24: Illustration of the simulation variables on the combined substrate + BDA system. The color coding is: Ir - light grey, Co - blue, oxide O - light red, BDA O - bright red, H - white and C - dark grey. When placed on the CoO film on Ir substrate, the BDA molecule was allowed to move along the lattice vectors (X and Y) of the surface as well as rotate in plane (θ). The $\theta = 0^\circ$ configuration has the molecule's axis along the [01-1] lattice direction. The lattice vectors of the surface are $(13.7, -2.7, 0.0)$ and $(0.0, 5.5, 0.0)$, while the simulation cell vector in y-direction is three times as long as the second lattice vector. The search space on the surface is one entire lattice unit cell (marked in green).

variables in the search for an adsorption site were the planar coordinates X and Y as well as the angle of rotation in plane θ (see Figure 24).

The simulation cell was here smaller than in case of the isolated molecule. Now the cell was spanned by vectors:

$$(13.7, -2.7, 0.0); (0.0, 16.5, 0.0); (0.0, 0.0, 19.4)$$

The lattice vectors of the Ir-CoO substrate which also define the search space for the BOSS searches are $(13.7, -2.7, 0.0)$ and $(0.0, 5.5, 0.0)$.

The energy value calculated was now the adsorption energy defined as $E_{ads} = E_{comb} - (E_{mol} + E_{surf})$. We calculated E_{mol} separately for each E_{ads} acquisition by doing a static simulation of the isolated molecule in the configuration as it was in the combined (molecule + substrate) simulation. We took this approach, because the relatively small simulation cell allows the molecule to significantly interact with the periodic copies of itself, so the molecule orientation in the cell affects the energy of the molecule-molecule interaction. E_{surf} was calculated only once and had the value $E_{surf} = -1126.070$ eV.

1D adsorption height search

We did the first 1D search in Z direction to determine the approximate adsorption height. The result only reveals the adsorption height at one location of the surface with one configuration of the molecule. However, the height is expected to be roughly similar for other locations and configurations too, because we expect dispersive interaction between molecule and surface. The resulting curve shows the expected Lennard-Jones behavior and an optimal height of 2.64Å (and $E_{ads} = -1.644$eV) for flat BDA molecule with $(X, Y, \theta) = (0.5, 0.5, 0.0)$ (see Figure 25 "flat" curve). The adsorption height is here defined as the distance in Z-direction between lowest atom of the molecule and highest atom of the surface (an oxygen).

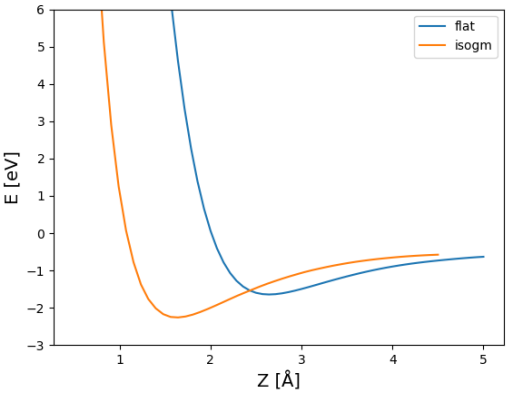


Figure 25: 1D searches in the Z direction with flat BDA at the center of search space and isolated global minimum BDA (isogm) at the predicted best adsorption site.

The orange curve in Figure 25 is a similar Z-search but repeated for the twisted (isolated minimum) BDA at the predicted best adsorption site $(X, Y, \theta) = (0.0, 0.2, 80.4)$,

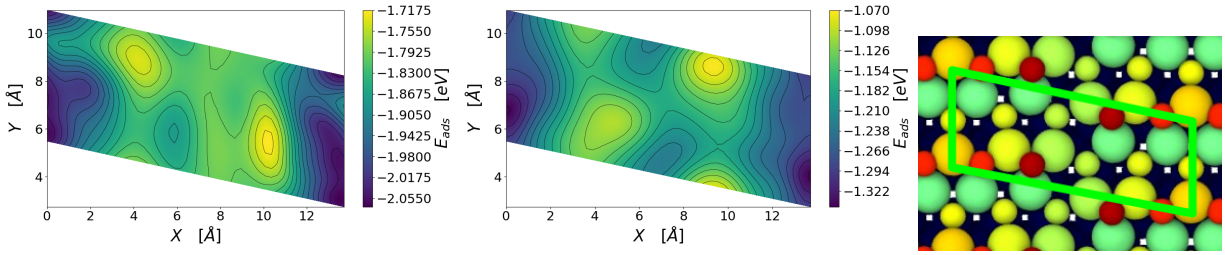
which we found later. Note that we did this search only after finding the final relaxed adsorption structures, so to be able to confirm the validity of the adsorption height the later results indicated. This search showed an adsorption height of 1.64Å with $E_{ads} = -2.260$eV. Thereby it confirmed that the molecule does adsorb significantly (about one Å) closer to the surface when it is twisted and placed in the optimal adsorption site.

The rest of the BOSS searches were conducted keeping the adsorption height fixed to 2.64Å. While this later turned out to be too high compared to relaxed adsorption structures, the assumption is that the energy landscapes of the other variables (X, Y, θ) are merely shifted or dampened by the high choice of Z but their structure is the same. Thus we will find the same adsorption sites as we would have with a slightly lower choice of Z, and the structural relaxations will then allow the molecule to move closer to the surface.

3D searches

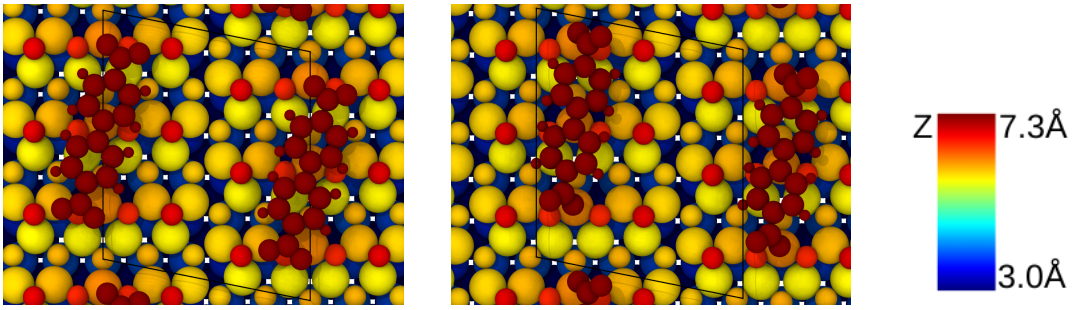
The actual adsorption site search was conducted for both BDA structures **flat** and **isogm** in 3D. We again fixed the molecule's internal structure but now the variables X, Y and θ were simultaneously optimized. We set the height to 2.64Å (distance from lowest atom in BDA to highest surface atom). Note that this means the molecular axis is at $Z = 2.64$Å for **flat** and at $Z = 3.41$Å for **isogm**.

We ran BOSS for a little over 200 iterations to produce converged models of the energy landscapes for both molecular structures on the Ir-CoO substrate. The resulting maps of the energy landscape in XY-plane are shown in Figure 26. The maps are cross-sections of the 3D landscape at best predicted value of θ. The global minimum adsorption configurations found in the BOSS search are shown in Figures 27 and 28a-b and their energies among other details are shown in Table 3.



(a) Flat XY-map 227 pts $\theta = 74.6°$ (b) Isogm XY-map 260 pts $\theta = 80.4°$ (c) Surface at search space

Figure 26: Models and XY-maps with predicted minimum value of the in plane rotation angle θ



(a) Flat (b) Isogm (c) Color scale

Figure 27: Predicted global minimum structures for the two molecular configurations shown on a periodically extended substrate. Color coded by height from $Z = 3.0$Å to $Z = 7.3$Å.

For both **flat** and **isogm** the predicted global minimum adsorption site is roughly the same: θ near the [011]-direction and $X \approx 0$. This configuration aligns the molecule with the row of the highest lying cobalt atoms (see Figure 29). The value of Y is slightly different for the two cases, but the maps in Figure 26 show that Y doesn't affect the energy much at $X \approx 0$. The energy scale and also the minimum adsorption energy lay higher for **isogm** (-1306.34eV) than for **flat** (-1306.68eV). As stated earlier this is caused by the molecular axis begin placed higher for **isogm** than for **flat**.

When $X \not\approx 0$, the maps have secondary minima approximately 0.1eV higher in energy than the global minima. These local minima are in different locations for the two cases, so they depend on the molecule's structure. A full list of these local minima are shown in appendix in Tables 6 and 7. While one could find other adsorption structures by exploring the local minima, we decided to extract only the global ones for further analysis and structural relaxation.

Structural relaxations

While BOSS reliably scans the entire parameter space for minimum energy structures, it is restricted to only vary the simulation variables we have chosen. For this reason a structural relaxation started from the end result of a BOSS search will decrease the energy even further, as all the previously fixed atoms are allowed to relax. We did this for both of the above adsorption structures (**flat** and **isogm**) using 400 iterations in VASP structural optimization. The average and maximum forces dropped down to values ($avg|F| = 0.008$eV/Å, $max|F| = 0.072$eV/Å) for **flat** and ($avg|F| = 0.008$eV/Å, $max|F| = 0.038$eV/Å) for **isogm**. This indicates the structures have been well relaxed.

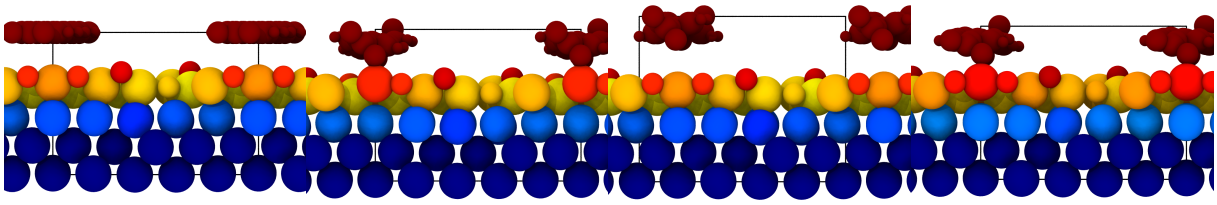
Structure	E_{TOT}[eV]	E_{ADS}[eV]	h_{BDA}[Å]	$O1$[°]	$O2$[°]	C[°]	ΔZ_{surf}[Å]	θ[°]
Flat BOSS	-1306.68	-2.07	2.6±0.0	0.0	0.0	0.0	0.00	74.6
Flat rlx	-1309.32	-4.34	2.3±1.0	-33.7	48.4	-16.6	0.06	82.0
Isogm BOSS	-1306.34	-1.34	3.4±0.8	-60.0	60.0	30.0	0.00	80.4
Isogm rlx	-1309.28	-4.36	2.4±1.0	-23.6	66.0	7.8	0.17	82.1

Table 3: Metrics of the found adsorption structures and their relaxed counterparts.

Images of the resulting relaxed adsorption structures are shown in Figures 30 and 31 and measured metrics in Table 3. The metrics not explained before are:

- h_{BDA} – height of the BDA molecular axis and average of largest deviation $(((h_{BDA} - h_{BDA}^{min}) + (h_{BDA}^{max} - h_{BDA}))/2)$
- ΔZ_{surf} – maximum change in the substrate atoms' heights compared to the isolated reference structure of the substrate

The h_{BDA} metric quantifies the absorption height measured roughly from the molecular axis and its variation (the ± value) how flat or twisted it is. The change in substrate atoms heights describes the response of the substrate to the adsorption: are substrate atoms moving higher to meet the molecule or deflecting away from it.



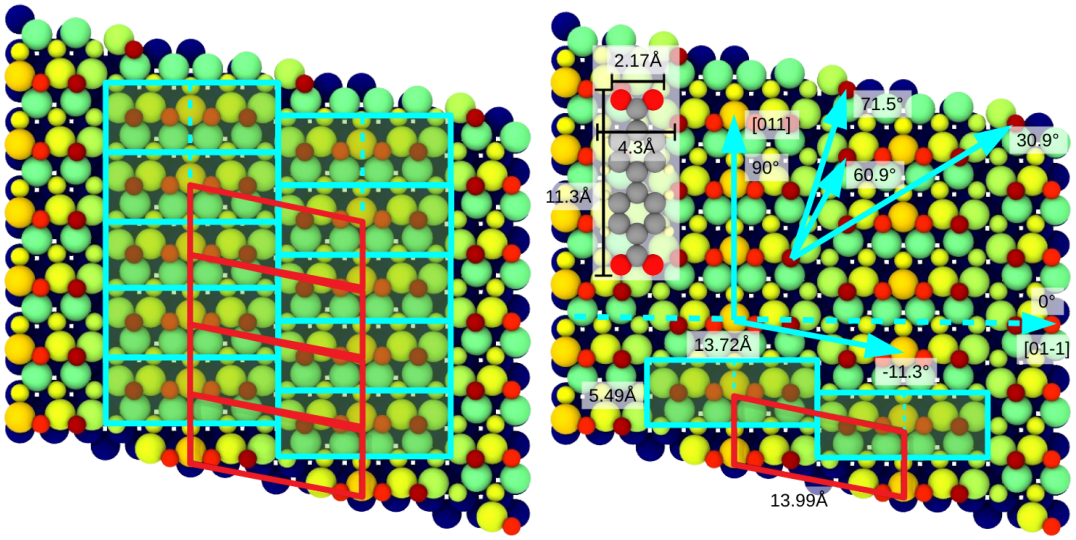
(a) Flat (b) Flat relaxed (c) Isogm (d) Isogm relaxed

Figure 28: Predicted global minimum structures for the two molecular configurations shown on a periodically extended substrate. Color coded by height (Z = 3.0Å→ Z = 7.3Å, see color scale in Figure 27).

It can be seen that in the relaxation, the adsorption positions (Xs and Ys) and molecule's planar angle (θs) are roughly maintained. What has considerably changed are the rotations of the functional groups and the adsorption height. The molecule which started flat (**flat**), has twisted its carboxyl groups so that one oxygen is closer to the substrate than the other ($O1 = -34°$ and $O2 = 48°$). In the other relaxation (**isogm**)

the other carboxyl group has bent even more (60°→ 66°) but the other one has straightened out somewhat (−60°→ −23°). The central angle between the benzene rings (C) has straightened out for **isogm** and bent only slightly for **flat**. This indicates that the BDA oxygens feel strong attraction to the surface while the other atoms do not.

Additionally the entire molecule has come closer to the surface in both relaxations. The surface atoms have significantly moved higher to meet the molecule only in case of the molecule which started already twisted (**isogm**). This can be seen in the ΔZ_{surf}-column in Table 3.



(a) Symmetries (b) Measurements

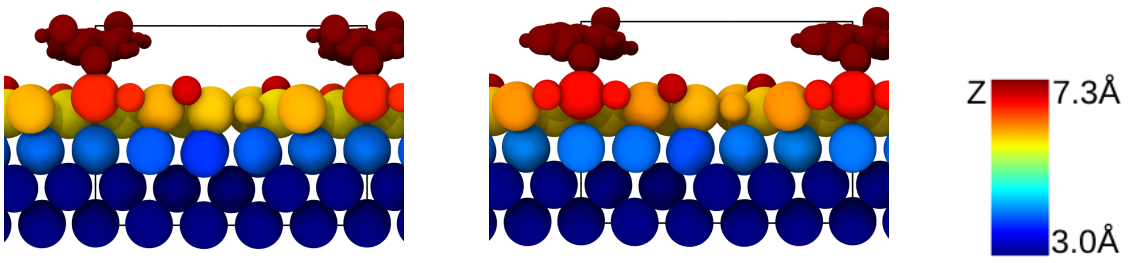
Figure 29: Illustrations of the substrate used in the analysis. Substrate extended periodically and color coded by height (Z = 3.0Å→ Z = 7.3Å, see color scale in Figure 27). On the right (a) the unit cell of the substrate structure is a tilted rectangle, but another symmetry – shown by the cyan rectangles – associated with the lattice directions of the underlying Ir(100)-surface can be identified. The dashed vertical lines indicate a mirror symmetry line for the cyan rectangles. On the left (b) various characteristic angles and distances for both the substrate and the BDA molecule are shown. They have been measured to be able to possibly relate the metrics of the found adsorption structures to characteristics of the substrate.

The twisted carboxyl groups and low height seem to enable bonding between the substrate cobalt atoms and the oxygens in the molecule. This hypothesis is supported firstly by the significant rise of the cobalt atoms below the carboxyl groups (seen best in Figures 30a-b and 31a-b.) and secondly by the molecular axis being aligned on top of the row of the highest lying cobalts (in [011]-direction). This row can be seen in the

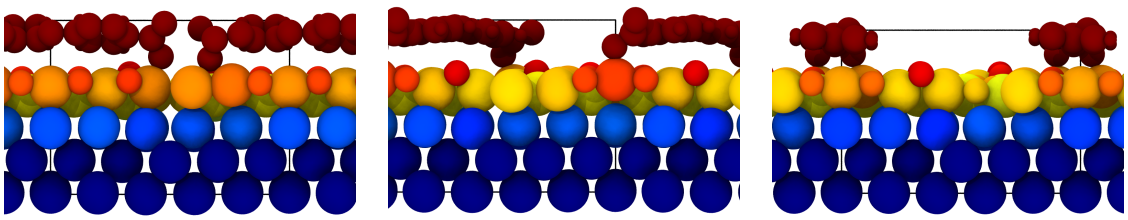
illustrations of the clean substrate in Figure 29 as a vertical line of the most orange (highest Z) cobalts.

4.3.4 Discussion and comparison

The tentative adsorption structures found before[3] are here called **BDAte-0**, **BDAte-45** and **BDAte-90** and shown in figures 30d-f and 31d-f. Their naming corresponds to the angle the BDA molecular axis is making with the $[01-1]$-direction: 0°, 45°, 90°. The binding energies for the three tried structures were found to be -0.60eV, -1.36eV and -1.43eV (respectively) after brief structural relaxation. The **BDAte-90**-structure had the lowest energy. Comparison of those adsorption configurations to the ones found in this study is shown visually in Figures 30 and 31 and analytically in Table 4. The previously identified structures show similar twisting angles of the functional groups in the molecule as were found in this study but the adsorption heights, sites and energies (see Table 4) are different. It is only the **BDAte-90**-structure that has the near-[011] alignment close to the high cobalt row, as found in **flat**- and **isogm**-structures. Nevertheless the surface atoms haven't elevated to bond with the BDA-oxygens even in the **BDAte-90**, which might explain the adsorption energy being higher by several eV. An alternative possibility is that **BDAte-0**, **BDAte-45** and **BDAte-90** may not be fully relaxed structures.

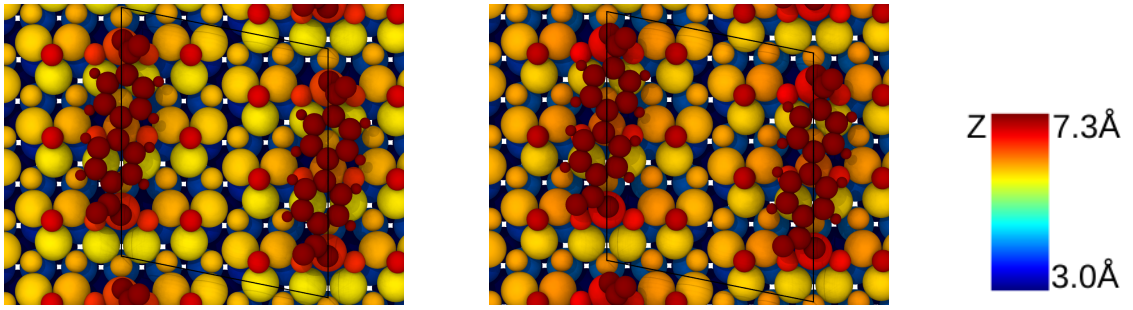


(a) Flat (b) Isogm (c) Color scale

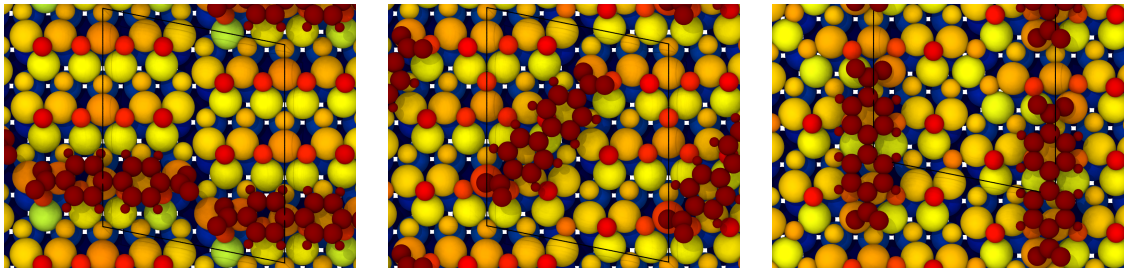


(d) BDAte-0 (e) BDAte-45 (f) BDAte-90

Figure 31: Side view of the five relaxed adsorption configurations to compare. Color coded by height ($Z = 3.0$Å$\rightarrow$ $Z = 7.3$Å—blue$\rightarrow$red).



(a) Flat (b) Isogm (c) Color scale



(d) BDAte-0 (e) BDAte-45 (f) BDAte-90

Figure 30: Top view of the five relaxed adsorption configurations to compare. Color coded by height ($Z = 3.0$Å$\rightarrow Z = 7.3$Å

Structure	E_{TOT}[eV]	E_{ADS}[eV]	h_{BDA}[Å]	$O1$[°]	$O2$[°]	C[°]	ΔZ_{surf}[Å]	θ[°]
Flat rlx	-1309.32	-4.34	2.3±1.0	-33.7	48.4	-16.6	0.06	82.0
Isogm rlx	-1309.28	-4.36	2.4±1.0	-23.6	66.0	7.8	0.17	82.1
BDA-0	-	-0.60	2.3±1.1	31.3	-36.3	0.7	0.02	0.6
BDA-45	-	-1.36	2.3±1.0	-37.4	38.2	7.3	0.01	48.2
BDA-90	-	-1.43	2.0±0.7	22.9	34.8	4.3	-0.02	86.9

Table 4: Comparison of the relevant structures' metrics

One could have logically gone further in the BOSS optimizations to perform a 7D (internal rotations + $XY\theta$ + rotation around molecular axis) search, but we felt that the current work followed by structural relaxations may have already captured the relevant chemistry in the system. We tried both flat and twisted molecule adsorption, but in both cases the relaxed configuration had fenyl rings fairly flat in surface plane and carboxyl groups twisted. Our study could be expanded by relaxing some of the found local minimum structures which didn't lie along the row of high cobalt atoms.

4.3.5 Conclusions

Two new adsorption configurations were found with adsorption energies (-4.34eV and -4.36eV) lower than in the previously found structures. The adsorption site and molecular orientation are similar in both new configurations and the **BDA-90** found before. We related the new structures' geometry to the chemistry between the substrate and the BDA molecule. The key factor in lowering the adsorption energy seems to be the bonding between surface cobalt atoms and one of the oxygens in the twisted carboxylic groups at each end of the BDA.

5 Conclusions

In this work I motivated and described an active machine learning based optimization method for atomistic structure search called BOSS. The theory and basic algorithm were presented and various improvements and alternative strategies discussed and illustrated. Finally I showed demonstrations of the efficiency and accuracy of the predictive power of BOSS. The demos consisted of three separate studies featuring benchmarking of the scaling of BOSS efficiency, a conformer search problem and a surface adsorption study with novel materials.

The BOSS method combines the Bayesian optimization algorithm with Gaussian process models and total energy simulations. BOSS does global atomistic structure search using chemical building blocks to reduce the search phase space to manageable dimensions. This way fewer total energy simulations are needed in mapping the minimum energy structures, which enables computational studies of large complex systems – such as organic/inorganic interfaces in heterogeneous devices. These systems are difficult to study experimentally and have so far been infeasible to study computationally due to costly simulations.

Alongside global optimization of the atomistic structure, BOSS produces a surrogate model of the energy landscape, which can reveal other possible local minimum structures and barriers between them. This allows one to make predictions about the dynamic behavior of the system without executing any dynamic simulations. For many systems of interest such simulations would not be feasible in required time scales using accurate *ab initio* simulation methods.

The dimensional scaling study using analytic functions (section 4.1) quantified the scaling of BOSS efficiency for fundamentally different functions when dimension increases. The study also benchmarked the slowdown in the scaling of efficiency when the optimized function had no periodicity or had correlation between the variables in higher dimensions.

The results also revealed the target function's derivative's important role to the optimization efficiency. The outcome will help people with choosing the simulation variables so that they are efficient to optimize, as well as help them estimate roughly how many BOSS iterations are potentially needed until convergence.

The predictive efficiency and accuracy of BOSS was showcased in the conformer search of alanine dipeptide molecule (section 4.2). The two most stable conformers and the characteristic 2D potential energy map was found with at least less than 10% of the effort of best alternative methods that we compared BOSS to. Also efficiency gained from the inclusion of gradient information in the GP model was found to be a very significant factor halving the number of simulations needed.

The value of BOSS in novel materials research was showcased in the surface adsorption study (section 4.3) of bifenyldicarboxylic acid on CoO thin film using DFT simulations. We found two adsorption configurations which had a lower energy than previous calculations and approximately supported the experimental data on the system.

The applications have shown that BOSS can significantly reduce the computational load of of atomistic structure search while maintaining predictive accuracy. It allows material scientists to study novel materials more efficiently, and thus help tailor the materials' properties to better suit the needs of modern devices. The code that was created in the process of this work to implement the BOSS method, will be made open-source for the scientific community to use. This way the project which lead to this work benefits everybody.

6 Glossary

A collection of short descriptions for important terms encountered repeatedly in this work.

- **Objective function** - The target function $f(\mathbf{x})$ which we wish to model and optimize. In structure search this is the total or potential energy as a function of the chosen variables.

- **Simulation variables** - The variables $\mathbf{x}$ as a function of which we consider the objective function $f(\mathbf{x})$ – i.e. the coordinates spanning the phase space one is optimizing. They could be for example rotations or translations of a molecule on a surface, internal rotations of bonds or bond angles of a molecule, or coordinates of an individual atom.

- **Data** (ensemble) - A finite collection of data $(\mathbf{x}, y) = (\mathbf{x}, f(\mathbf{x}))$ sampled from the objective function. If one is using gradient observations, the data ensemble will keep a record of them too $(\mathbf{x}, f(\mathbf{x}), \partial_{\mathbf{x}} f(\mathbf{x}))$.

- **Bayesian optimization** BO - A strategy to treat the objective function as an unknown function, and given an ensemble of objective function evaluations, update a prior into a posterior distribution (the GP model) over the objective function domain. The strategy is iterative, because an acquisition function is constructed based on the posterior distribution, to determine the next query point (sampling location).

- **Gaussian process** GP - The joint Gaussian distribution of random variables indexed by a continuous space. Given training data and a kernel function, a GP model can be used to fit the most likely surrogate model on the data. The predictions made using the surrogate model are one dimensional normal distributions, so they provide a measure of uncertainty (standard deviation) in addition to the predicted value (mean) at a given point in space.

- **Acquisition function** - A function to determine the next query point (or sampling location) for data collection in Bayesian optimization strategy. Acquisition function is constructed based on the GP model and it tries to balance between exploiting the predicted minimum locations and exploring the space.

- **Kernel function** (covariance function) - A measure of similarity between data points in the data ensemble. There exist different kinds of kernels, and they have a varying number of free parameters called hyperparameters.

- **Hyperparameters** - The free parameters of the kernel function, which affect the shape of the GP model. Common hyperparameters are variance, length scale and period. Hyperparameters are sometimes optimized to maximize the marginal likelihood of the GP model fit – i.e. give out the best fit to the available data.

7 Appendix

A collection of additional data left out of the main text in order to keep it compact.

d4 [°]	d8 [°]	d10 [°]	d12 [°]	d16 [°]	d18 [°]	d20 [°]	energy [kcal/mol]
Default values							
180.0	120.0	0.0	180.0	240.0	0.0	0.0	-
Lowest local minima from 7D BOSS search							
11.9	144.1	0.4	8.5	219.2	-0.2	10.2	-28.6
8.9	213.3	3.3	5.2	132.3	1.8	29.5	-26.4
9.4	212.5	4.4	5.7	127.0	2.3	-50.0	-26.2
14.5	158.4	1.6	10.5	82.1	0.4	5.6	-25.9
9.3	212.7	4.3	70.0	127.7	2.3	-50.0	-25.7
-50.0	163.0	0.2	10.6	71.9	1.5	6.1	-20.8

Table 5: Default values and optimal values for alanine dipeptide molecule's dihedral angles. The optimal values and the corresponding energies are shown in six lowest (limited to < -20kcal/mol) local minima found in the 7D BOSS search in section 4.2. The 2.-5. minima are all essentially the same minimum as they mainly vary by the methyl groups' positions which matter very little to energy.

X [frac. $\hat{a}$]	Y [frac. $\hat{b}$]	Θ [°]	E_{ADS} [eV]
0.988	0.054	74.612	-2.070
0.987	0.387	69.474	-2.067
0.008	0.104	104.965	-2.063
0.008	0.102	104.787	-2.063
0.891	0.292	94.037	-2.060
0.890	0.291	94.451	-2.060
0.026	0.516	110.623	-2.057
0.908	0.575	83.927	-2.045
0.906	0.579	85.484	-2.045
0.121	0.467	83.049	-2.034

Table 6: Ten lowest local minima found in the 3D BOSS search of **flat** BDA molecule adsorption on the CoO thin film on Ir substrate. Θ is the in plane rotation of the molecular axis, while X and Y are fractions of the corresponding surface unit vectors and determine the location of the center of the molecule. The lowest molecule atom to highest surface atom distance is fixed to 2.64Å.

X [frac. $\hat{a}$]	Y [frac. $\hat{b}$]	Θ [°]	E_{ADS} [eV]
0.004	0.237	80.437	-1.343
0.004	0.238	80.208	-1.343
0.093	0.388	68.281	-1.309
0.093	0.387	68.460	-1.309
0.094	0.389	68.087	-1.309
0.735	0.889	121.109	-1.297
0.979	0.641	73.154	-1.290
0.979	0.641	73.177	-1.290
0.670	0.440	27.893	-1.283
0.244	0.381	114.081	-1.283

Table 7: Ten lowest local minima found in the 3D BOSS search of twisted (**isogm**) BDA molecule adsorption on the CoO thin film on Ir substrate. Θ is the in plane rotation of the molecular axis, while X and Y are fractions of the corresponding surface unit vectors and determine the location of the center of the molecule. The lowest molecule atom to highest surface atom distance is fixed to 2.64Å. This makes the molecular axis lie higher than that for the twisted molecule.

References

[1] Amber. http://ambermd.org/. Accessed: 02-07-2018.

[2] Open Babel: The Open Source Chemistry Toolbox. http://openbabel.org/wiki/Main_Page. Accessed: 02-07-2018.

[3] Private communication - A. Schneider.

[4] Private communication - M. Todorović.

[5] VASP - Vienna Ab initio Simulation Package. https://www.vasp.at/. Accessed: 22-01-2019.

[6] Veronika Brázdová, David R Bowler, and P Harrison. *Atomistic Computer Simulations*. 2012.

[7] M. Reuschl L. Hammer C. Tröppner, T. Schmitt and M. A. Schneider. Incommensurate Moiré overlayer with strong local binding: CoO(111) bilayer on Ir(100). *Phys. Rev. B 86, 235407*, 2012.

[8] Eliodoro Chiavazzo, Ronald R. Coifman, Roberto Covino, C. William Gear, Anastasia S. Georgiou, Gerhard Hummer, and Ioannis G. Kevrekidis. Intrinsic Map Dynamics exploration for uncharted effective free energy landscapes. *PNAS*, 114:E5494–E5503, 2017.

[9] Alexander Denzel and Johannes Kästner. Gaussian process regression for geometry optimization. *The Journal of Chemical Physics*, 2018.

[10] Thomas A. Frewen, Gerhard Hummer, and Ioannis G. Kevrekidis. Exploration of effective potential landscapes using coarse reverse integration. *Journal of Chemical Physics*, 2009.

[11] Michael U. Gutmann and Jukka Corander. Bayesian Optimization for Likelihood-Free Inference of Simulator-Based Statistical Models. 2015.

[12] P. Hohenberg and W. Kohn. Inhomogeneous Electron Gas. *Phys. Rev. 136, B864*, 1964.

[13] K. Burke J. P. Perdew and M. Ernzerhof. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett. 77, 3865*, 1996.

[14] W. Kohn and L. J. Sham. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev. 140, A1133*, 1965.

[15] J. P. Perdew and A. Zunger. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B 23, 5048*, 1981.

[16] Filippo Pullara and Ignacio J. General. Population reversal driven by unrestrained interactions in molecular dynamics simulations: A dialanine model. *AIP Advances*, 2015.

[17] C E Rasmussen, C K I Williams, Richard S Sutton, Andrew G Barto, Peter Spirtes, Clark Glymour, Richard Scheines, Bernhard Schölkopf, and Alexander J Smola. *Gaussian Processes for Machine Learning*. MIT Press MIT Press, 2006.

[18] SheffieldML. GPy. `https://github.com/SheffieldML/GPy`. Accessed: 02-07-2018.

[19] Eero Siivola. GPYgradients. `https://github.com/esiivola/GPYgradients/tree/feature_gradients`. Accessed: 02-07-2018.

[20] Paul E. Smith. The alanine dipeptide free energy surface in solution. *Journal of Chemical Physics*, 1999.

[21] I.M. Sobol. Distribution of points in a cube and approximate evaluation of integrals. *U.S.S.R Comput. Maths. Math. Phys. 7: 86–112*, 1967.

[22] E Solak, R Murray-Smith, W.E. Leithead, D.J. Leith, and C.E. Rasmussen. Derivative observations in Gaussian process models of dynamic systems. *Nips 15*, page 8, 2002.

[23] Birgit Strodel and David J. Wales. Free energy surfaces from an extended harmonic superposition approach and kinetics for alanine dipeptide. *Chemical Physics Letters*, 2008.

[24] K. H. Sutherland-Cash, D. J. Wales, and D. Chakrabarti. Free energy basin-hopping. *Chemical Physics Letters*, 2015.

[25] Douglas J Tobias and Charles L Brooks III. Conformational Equilibrium in the Alanine Dipeptide in the Gas Phase and Aqueous Solution: A Comparison of Theoretical Results. *J. Phys. Chem*, 96:3864–3870, 1992.